NASA CR-66818

N69-36472

FINAL REPORT

SIMULATED TRAJECTORIES ERROR ANALYSIS PROGRAM

VOLUME I: USER'S MANUAL

By Gentry Lee, Ralph Falce, Dr. Doyle Vogt,
Shearon Pearson, and Eva Demlow

Prepared under Contract No. NAS1-8745 by
MARTIN MARIETTA CORPORATION
Denver, Colorado

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Langley Research Center
Hampton, Virginia

FOREWORD

This document represents Volume I of the final report on NASA Langley Contract NAS1-8745 entitled "Simulated Trajectories Error Analysis Program". The report is prepared in two volumes:

Volume I - User's Manual;

Volume II - Analytical Manual.

CONTENTS

FINAL REPORT

SIMULATED TRAJECTORIES ERROR ANALYSIS PROGRAM

VOLUME I: USER'S MANUAL

By Gentry Lee, Ralph Falce, Dr. Doyle Vogt,
Shearon Pearson, and Eva Demlow

SUMMARY

This document presents the formulation, computational logic, input/output options, subroutine description and other pertinent information that should aid the user of the Simulated Trajectories Error Analysis Program (STEAP).

The program has four modes of operation -- trajectory, targeting, error analysis, and simulation. The trajectory and targeting modes are based on an n-body subroutine that uses the virtual mass or varicentric technique. The fundamental technique used in developing the error analysis and simulation modes is the Kalman recursive filter algorithm.

I. INTRODUCTION

This document is concerned with the design and implementation of a digital computer program to facilitate the study of interplanetary trajectories. This volume is a companion to Volume II (analytical manual) of the final report under NASA-Langley Contract NAS1-8745.

The Simulated Trajectory Error Analysis Program (STEAP) is a Fortran program that was written and checked out using the CDC-6400/6500 digital computer. To minimize possible system incompatabilities, care has been exercised to assure that only the basic features of the system are used. Thus, the program should be operable on most CDC equipment.

The program is divided into four operational modes. The first mode, trajectory mode, generates an n-body interplanetary trajectory using virtual mass concepts. The second mode determines necessary injection conditions for targeting to certain constraints near a target planet and is designated the targeting mode. The third operational mode is called the error analysis mode. The

primary function of this mode is to provide the capability for error studies with varying statistical input. Various measurement data are obtained and processed through an optimum recursive filter. The fourth mode in which the program can be exercised is the simulation mode. This mode tests the convergence of the orbit determination process and has the capability to simulate midcourse corrections.

The remaining chapters of this document discuss the various input/output options, main program structure, subroutine descriptions, flow charts and example runs.

II. INPUT OPTIONS

THIS SECTION CONTAINS A COMPLETE DESCRIPTION OF THE INPUT NECESSARY FOR EACH OF THE MODES OF OPERATION IN ADDITION TO ANY OPTIONS AVAILABLE.

IF THE TARGETING MODE IS TO BE USED PROCEED DIRECTLY TO PART B. THE FOLLOWING COMMENTS ARE APPLICABLE TO THE REMAINING MODES.

THE FIRST CARD WHICH SHOULD BE READ CONTAINS THE VARIABLE IRUNX WHICH INDICATES HOW MANY DIFFERENT RUNS ARE TO BE MADE. THIS NUMBER SHOULD BE PLACED ON THE CARD ACCORDING TO AN I10 FIELD. --NOTE--THIS NUMBER IS ONLY READ ONCE EACH TIME THE PROGRAM IS INPUT.

THE NEXT CARD SHOULD CONTAIN THE VARIABLES IPRO AND ITR WHICH GIVE THE PROBLEM IDENTIFICATION AND A CODE NUMBER INDICATING WHICH MODE TO BE USED FOR THIS PROBLEM. THE FORMAT IS 2I10. --NOTE--THIS CARD SHOULD PRECEDE THE INPUT FOR EACH RUN.

```
ITR      --  =1  --  TRAJECTORY MODE
             =3  --  ERROR ANALYSIS MODE
             =4  --  SIMULATION MODE
```

A. IF THE TRAJECTORY MODE IS TO BE RUN, THE DATA IS INPUT THROUGH THE USE OF A NAMELIST ENTITLED TRAJ WHICH INCLUDES THE FOLLOWING VARIABLES.

```
XI       --  A VECTOR CONTAINING THE INITIAL POSITION AND
             VELOCITY OF THE VEHICLE
             --NOTE--THIS VECTOR IS READ ONLY IF ICOOR = 0,1,2
ICOOR    --  A CODE TO DETERMINE WHAT COORDINATE SYSTEM THE
             INITIAL STATE VECTOR IS IN.  IF THIS CODE IS NOT
             INCLUDED IN THE NAMELIST, IT IS ASSUMED TO BE 2.
             =0 -- HELIOCENTRIC ECLIPTIC
             =1 -- GEOCENTRIC EQUATORIAL
             =2 -- GEOCENTRIC ECLIPTIC
             =3 -- JPL CONDITIONS (RDS,PHI,THETA,VEL,GAMMA,
                   SIGMA)
RDS      --  EARTH-CENTERED INJECTION RADIUS
PHI      --  DECLINATION
THETA    --  INJECTION RIGHT ASCENSION
VEL      --  INERTIAL INJECTION SPEED
GAMMA    --  INJECTION PATH ANGLE
SIGMA    --  INJECTION AZIMUTH
             --NOTE--THESE CONDITIONS ARE INPUT ONLY IF
             ICOOR = 3.
LMO      --  LAUNCH MONTH (INTEGER)
LDAY     --  LAUNCH DAYS (INTEGER)
LHR      --  LAUNCH HOURS (INTEGER)
LMIN     --  LAUNCH MINUTES (INTEGER)
SECL     --  LAUNCH SECONDS (FLOATING)
LYR      --  LAUNCH YEAR (INTEGER)
```

```
IMO      --  MONTH OF FINAL COMPUTATION (INTEGER)
IDAY     --  DAY OF FINAL COMPUTATION (INTEGER)
IHR      --  HOUR OF FINAL COMPUTATION (INTEGER)
IMIN     --  MINUTE OF FINAL COMPUTATION (INTEGER)
SECI     --  SECOND OF FINAL COMPUTATION (FLOATING)
IYR      --  YEAR OF FINAL COMPUTATION (INTEGER)
ALNGTH   --  LENGTH UNITS PER A.U.
             --NOTE--IF ALNGTH IS NOT READ IN, THE LENGTH UNITS
             ARE ASSUMED TO BE KILOMETERS (ALNGTH=149598500.)
TM       --  TIME UNITS PER DAY
             --NOTE--IF TM IS NOT READ IN, THE TIME UNITS ARE
             ASSUMED TO BE SECONDS. (TM=86400.)
NTMC     --  NOMINAL TRAJECTORY MODULE CODE
             =1 -- PATCHED CONIC (NOT SUPPLIED WITH THIS PROGRAM)
             =2 -- VIRTUAL MASS
             --NOTE--IF NOT INPUT, CODE 2 IS ASSUMED
TRTM1    --  INITIAL TRAJECTORY TIME (ASSUMED ZERO IF NOT
             READ IN)
NBOD     --  NUMBER OF BODIES TO BE CONSIDERED IN ANALYSIS
             --NOTE--IF NBOD IS NOT INPUT, IT IS ASSUMED TO BE 3
             (SUN, LAUNCH PLANET, TARGET PLANET)
NB       --  ARRAY OF CODES OF BODIES
             =1 -- SUN
             =2 -- MERCURY
             =3 -- VENUS
             =4 -- EARTH
             =5 -- MARS
             =6 -- JUPITER
             =7 -- SATURN
             =8 -- URANUS
             =9 -- NEPTUNE
             =10-- PLUTO
             =11-- EARTHS MOON
             --NOTE--NB(2)=LAUNCH PLANET
                     NB(3)=TARGET PLANET
THE FOLLOWING INFORMATION IS NECESSARY ONLY IF NTMC=2.
   IEPHEM --  EPHEMERIS CODE
             =0  --  PLACE EACH PLANET IN ELLIPSE
                     THE DATE AT WHICH THIS ELLIPSE IS
                     CALCULATED IS DETERMINED BY READING IN A
                     VARIABLE WHICH IS ENTITLED AS THE NAME OF
                     THE PLANET CONSIDERED.  THIS VARIABLE
                     SHOULD CONTAIN SIX NUMBERS SPECIFYING THE
                     MONTH, DAY, HOUR, MINUTE,SECOND, AND
                     YEAR.  (EXAMPLE..EARTH=7,24,6,15,38,1973)
                     --NOTE--IF THESE VARIABLES ARE OMITTED FROM
                     THE NAMELIST THE FOLLOWING RULES WILL BE
                     APPLIED.
                     NB(2) AT LAUNCH DATE
                     NB(3) AT TARGET DATE
                     MOON AT SAME DATE AS EARTH
                     ALL OTHERS AT TARGET DATE
             =1  --  CALCULATE ORBITAL ELEMENTS FOR EACH PLANET
                     EACH TIME INTERVAL
             --NOTE--ONE  IS ASSUMED IF OMITTED FROM INPUT
```

```
IPRINT   --  PRINT CODE FOR VIRTUAL MASS:
             =0  ALL OUTPUT WILL BE PRINTED BOTH INITIALLY AND
                 FINALLY AS USUAL
                 (ASSUMED IF NOT INPUT IN TRAJECTORY MODE)
             =1  OUTPUT WILL BE SUPPRESSED AT THE BEGINNING AND
                 FINAL STEPS
                 (ASSUMED IF NOT INPUT IN ERROR ANALYSIS MODE
                 AND SIMULATION MODE)
ISP2'    --  CODE FOR VIRTUAL MASS TRAJECTORY
             =0      CONTINUE INTEGRATING TO FINAL TIME
                     --NOTE--ISP2 IS ASSUMED ZERO IF NOT INPUT
             .GT.0   STOP INTEGRATING WHEN SPHERE OF INFLUENCE
                     OF TARGET PLANET IS ENCOUNTERED.
ACC      --  ACCURACY FIGURE  (ASSUMED 1.E-6 IF NOT INPUT)
DELTP    --  PRINT INTERVAL (IN DAYS)--ASSUMED 3. IF NOT INPUT
             IN TRAJECTORY MODE.  IF NOT INPUT IN ERROR ANALYSIS
             MODE OR SIMULATION MODE DELTP IS ASSUMED 1.E50.
INPR     --  PRINT INTERVAL (INCREMENTS) (ASSUMED 100 IF NOT
             INPUT IN TRAJECTORY MODE.  IN ERROR ANALYSIS OR
             SIMULATION MODE ASSUMED 7777777 IF NOT INPUT)
```

B. THE ENTIRE INPUT REQUIRED BY THE TARGETING PROGRAM IS SUPPLIED IN SIX CARDS WITH EACH INDIVIDUAL CARD CONTAINING UNIFIED DATA. THE (SEQUENTIAL) CARDS AND THEIR REQUISITE FORMAT ARE LISTED BELOW.

```
CARD 1 -- IDAT1(5),S1,IDAT2(5),S2
   FORMAT (I5,4I3,F7.3,5X,I4,4I3,F7.3)
CARD 2 -- NBOD, NB(NBOD)
   FORMAT (I2,3X,11I5)
CARD 3 -- INJEK, RS(6)
   FORMAT (I2,2X,3E15.8,3E10.3)
CARD 4 -- ITARG, TARG1, TARG2, TOL1, TOL2, TOL3
   FORMAT (I2,2X,5F15.5)
CARD 5 -- ISKEJ, AC(ISKEJ)
   FORMAT (I2,6X,7F10.8)
CARD 6 -- NITS, INCPR, TIMPR, BDELV
   FORMAT (I2,7X,I5,5X,F9.4,F11.8)
```

THE DEFINITIONS OF THE ABOVE DATA ARE SUMMARIZED BELOW.

```
IDAT1,S1 -  THE INITIAL TIME.  IDAT1 IS A 5-VECTOR COMPOSED OF
            THE INITIAL YEAR,MONTH,DAY,HOUR,AND MINUTE.  S1 DE-
            NOTES THE SECONDS.  IF INJEK=1, THIS TIME IS SPECI-
            FIED ONLY TO THE DAY. IF INJEK=2, THE TIME SHOULD
            BE PRESCRIBED TO THE NEAREST THOUSANDTH-SECOND.
IDAT2,S2 -  THE TARGET TIME.  IF ITARG=1,2,5,6 THIS IS THE TIME
            AT CLOSEST APPROACH OF THE TARGET PLANET.  IF ITARG
            =3,4  THIS IS THE TIME AT SPHERE OF INFLUENCE OF
            THE TARGET PLANET.
NBOD     -- THE NUMBER OF GRAVITATIONAL BODIES TO BE CONSIDERED
            IN THE INTEGRATION.
NB       -- A VECTOR OF DIMENSION NBOD SPECIFYING THE INDICES
            OF THE GRAVITATIONAL BODIES.  THE SECOND BODY IS
            ASSUMED TO BE THE LAUNCH PLANET, THE THIRD, THE
```

TARGET PLANET. THE NUMBERING SYSTEM ASSIGNS THE INDEX 1 TO THE SUN, 2 TO MERCURY, 3 TO VENUS, 4 TO EARTH, 5 TO MARS, 6 TO JUPITER, 7 TO SATURN, 8 TO URANUS, 9 TO NEPTUNE, 10 TO PLUTO, AND 11 TO THE EARTHS MOON.

INJEK -- A FLAG DESIGNATING WHICH OF TWO INJECTION OPTIONS IS TO BE USED.

IF INJEK = 1 - THE POINT-TO-POINT CONDITIONS ARE TO BE COMPUTED AND USED AS THE ZERO ITERATE INJECTION CONDITIONS.

= 2 - THE ZERO ITERATE INJECTION CONDITIONS ARE READ IN.

RS -- THE ZERO ITERATE INJECTION POSITION AND VELOCITY IN HELIOCENTRIC ECLIPTIC COORDINATES. IF INJEK = 1, THE CORRESPONDING COLUMNS ARE LEFT BLANK.

ITARG -- A FLAG DESIGNATING WHICH OF SIX TARGET OPTIONS ARE TO BE IN EFFECT. THE OPTIONS ARE

ITARG	OPTION
1	POINT-TO-POINT CONDITIONS
2	PATCHED CONIC CONDITIONS (UNBIASED PTP)
3	B.T, B.R, APPROXIMATE TSI
4	B.T, B.R, TSI
5	APPROXIMATE RCA, ICA, TCA
6	EXACT RCA, ICA, TCA

TARG1, TARG2 - TARGET PARAMETERS. THE PARAMETERS HAVE THE FOLLOWING DEFINITIONS DEPENDING ON THE TARGET OPTION.

ITARG	TARG1	TARG2
1,2	DO NOT APPLY	
3,4	B.T (KM)	B.R (KM)
5,6	INC (DEG)	RCA (KM)

TOL1, TOL2, TOL3 -- TARGET TOLERANCES. THE TOLERANCES SPECIFY THE ERROR THAT WILL BE ACCEPTABLE IN THE TARGET PARAMETERS ACCORDING TO THE FOLLOWING SCHEME

ITARG	TOL1	TOL2	TOL3
1,2		DO NOT APPLY	
3,4,5	B.T (KM)	B.R (KM)	TSI (DAYS)
6	INC (DEG)	RCA (KM)	TCA (DAYS)

ISKEJ -- A FLAG DESIGNATING THE NUMBER OF ACCURACY LEVELS TO BE USED IN THE TARGETING PROCESS.

AC -- A VECTOR OF DIMENSION ISKEJ WHOSE COMPONENTS ARE THE PROGRESSIVE ACCURACY LEVELS FROM THE LOWEST TO THE DESIRED FINAL LEVEL.

NITS -- THE MAXIMUM NUMBER OF ITERATIONS ALLOWED AT THE FINAL ACCURACY LEVEL.

INCPR -- THE NUMBER OF INTEGRATION INCREMENTS BETWEEN EACH PRINTOUT OF TRAJECTORY INFORMATION IN THE FINAL INTEGRATION OF THE TARGETED INJECTION CONDITIONS.

TIMPR -- THE NUMBER OF DAYS BETWEEN EACH PRINTOUT OF TRAJECTORY INFORMATION IN THE FINAL INTEGRATION OF THE TARGETED INJECTION CONDITIONS.

BDELV -- THE BASIC VELOCITY INCREMENT BY WHICH THE NOMINAL VELOCITIES ARE PERTURBED IN COMPUTING STATE TRANSITION MATRICES. IN OUTER TARGETING THE VELOCITY INCREMENT IS 10 TIMES GREATER, IN CLOSEST APPROACH TARGETING IT IS 1/10 AS LARGE.

C. IF THE ERROR ANALYSIS MODE IS TO BE RUN, A NAMELIST ENTITLED ERRAN IS READ WHICH INCLUDES ALL OF THOSE VARIABLES USED IN THE TRAJECTORY MODE PLUS THE FOLLOWING.

```
IAUG      --  AUGMENTATION FLAG
              = 1    STATE VECTOR CONSISTS OF POSITION AND
                     VELOCITY OF VEHICLE (NDIM = 6)

          --  ALL REMAINING CODES INDICATE STATE VECTORS WITH
              AUGMENTED INFORMATION AS NOTED  --

              = 2    STATION LOCATION PARAMETERS (NDIM = 9)
                     (GEOCENTRIC RADIUS, LATITUDE, LONGITUDE)
              = 3    MU OF SUN AND MU OF TARGET PLANET (NDIM = 8)
              = 4    SIX MEASUREMENT BIASES (RANGE BIAS, RANGE-
                     RATE BIAS, THREE STAR ANGLE BIASES, APPARENT
                     PLANET DIAMETER BIAS)  (NDIM=12)
              = 5    THREE EPHEMERIS BIASES OF TARGET PLANET
                     (SEMI-MAJOR AXIS BIAS, ECCENTRICITY BIAS,
                     INCLINATION BIAS)  (NDIM=9)
              = 6    NINE STATION LOCATION PARAMETERS (NDIM=15)
                     (THREE FROM EACH OF THREE STATIONS)
              = 7    THREE STATION LOCATION PARAMETERS PLUS MU OF
                     SUN AND MU OF TARGET PLANET (NDIM = 11)
              = 8    THREE STATION LOCATION PARAMETERS AND SIX
                     MEASUREMENT BIASES  (NDIM=15)
              = 9    MU OF SUN, MU OF TARGET PLANET, THREE
                     EPHEMERIS BIASES  (NDIM=11)
              = 10   SIX MEASUREMENT BIASES AND THREE EPHEMERIS
                     BIASES  (NDIM=15)
              = 11   THREE STATION LOCATION PARAMETERS PLUS MU
                     OF SUN, MU OF PLANET, SIX MEASUREMENT
                     BIASES (NDIM=17)

NENT      --  NUMBER OF ENTRIES IN THE MEASUREMENT SCHEDULE
              THIS IS ASSUMED ZERO IF IT IS NOT INPUT
              --NOTE--THE MEASUREMENT SCHEDULE ITSELF IS NOT
              READ IN THE NAMELIST.  IT WILL BE READ
              IMMEDIATLEY FOLLOWING THE NAMELIST
              SCHED(1)    SCHED(2)    SCHED(3)    MEAS
                F10.0       F10.0       F10.0      I10
              DAY1     TO  DAY2  EVERY X DAYS W. CODE
              THE CODE IS DETERMINED BY THE FOLLOWING LIST
              =1  RANGE-RATE --  IDEALIZED STATION
              =2  RANGE, RANGE-RATE -- IDEALIZED STATION
              =3  RANGE-RATE -- STATION 1
              =4  RANGE, RANGE-RATE --  STATION 1
              =5  RANGE-RATE -- STATION 2
              =6  RANGE, RANGE-RATE --  STATION 2
              =7  RANGE-RATE -- STATION 3
              =8  RANGE, RANGE-RATE --  STATION 3
              =9  THREE STAR PLANET ANGLES
              =10 APPARENT PLANET DIAMETER
NEV1      --  NUMBER OF EIGENVECTOR EVENTS (ASSUMED ZERO, IF
              NOT READ)
```

```
T1       --  ARRAY OF TIMES AT WHICH EIGENVECTOR EVENTS OCCUR
             --NOTE--THIS IS TO BE INPUT ONLY IF NEV1.NE.0
IEIG     --  EIGENVECTOR CODE
             =0 -- ONLY POSITION EIGENVECTORS WILL BE INPUT
                  (ASSUMED IF NOT INPUT)
             =1 -- BOTH POSITION AND VELOCITY EIGENVECTORS
                  WILL BE CALCULATED
IHYP1    --  HYPERELLPSOID SIGMA LEVEL CODE
             =1 -- SIGMA LEVEL EQUALS ONE
             =2  - SIGMA LEVEL EQUALS THREE (ASSUMED IF NOT
                  INPUT)
             =3 -- SIGMA LEVEL OF BOTH ONE AND THREE
NEV2     --  NUMBER OF PREDICTION EVENTS (ASSUMED ZERO, IF
             NOT READ)
T2       --  ARRAY OF TIMES AT WHICH PREDICTION EVENTS OCCUR
             --NOTE--THIS IS INPUT ONLY IF NEV2.NE.0
TPT2     --  ARRAY OF TIMES TO WHICH ONE WISHES TO PREDICT.
             --NOTE--THESE MUST CORRESPOND TO THOSE TIMES
             LISTED IN T2 AND SHOULD BE INPUT ONLY IF T2 IS
             INPUT
NEV3     --  NUMBER OF GUIDANCE EVENTS (ASSUMED TO BE ZERO IF
             NOT INPUT)
T3       --  ARRAY OF TIMES AT WHICH GUIDANCE EVENTS OCCUR
             --NOTE--THESE MUST BE INPUT ONLY IF NEV3.NE.0
ICDT3    --  ARRAY OF CODES WHICH DETERMINE WHAT GUIDANCE
             POLICY IS TO BE USED AT EACH GUIDANCE EVENT
             =1 -- FIXED TIME OF ARRIVAL
             =2 -- TWO-VARIABLE B-PLANE
             =3 -- THREE-VARIABLE B-PLANE
             --NOTE--THESE CODES MUST CORRESPOND TO THE TIMES
             AS STATED IN T3 AND NEED BE INPUT ONLY IF T3 IS
             INPUT--IF THESE ARE NOT INPUT WHEN T3 IS INPUT
             THREE VARIABLE B-PLANE IS ASSUMED
ICDQ3    --  ARRAY OF CODES FOR GUIDANCE EVENTS TO DETERMINE
             HOW THE EXECUTION ERROR IS TO BE CALCULATED.
             = 0  CALCULATED DIRECTLY FROM S MATRIX
             = 1  CALCULATED FROM EIGENVECTOR CORRESPONDING TO
                  MAXIMUM EIGENVALUE OF S MATRIX.
             --NOTE--THESE CODES MUST CORRESPOND TO THE TIMES AS
             STATED IN T3 AND NEED BE INPUT ONLY IF T3 IS INPUT.
             IF THESE ARE NOT INPUT WHEN T3 IS INPUT, OPTION 1
             IS ASSUMED
SIGRES   --  VARIANCE OF RESOLUTION ERROR
             ASSUMED 4.E-8 KM**2/SEC**2 IF NOT INPUT
SIGPRO   --  VARIANCE OF PROPORTIONALITY ERROR
             ASSUMED .0001 IF NOT INPUT
SIGALP   --  VARIANCE OF POINTING ANGLE ALPHA
             ASSUMED .0043625 RADIANS IF NOT INPUT
SIGBET   --  VARIANCE OF POINTING ANGLE BETA
             ASSUMED .0043625 RADIANS IF NOT INPUT
             --NOTE--THE ABOVE SIGMA VALUES MUST BE INPUT ONLY
             IF NEV3.NE.0
```

```
P        --  INITIAL COVARIANCE MATRIX.
             --NOTE--IF THIS MATRIX IS NOT INPUT, A DIAGONAL
             MATRIX IS ASSUMED WITH THE LISTED VALUES FOR THE
             FIRST SIX ELEMENTS ON THE DIAGONAL.  ALL OTHERS
             WILL BE ZERO
             1.     1.     1.     1.E-4     1.E-4     1.E-4
ISTMC    --  STATE TRANSITION MATRIX CODE
             =1 -- PATCHED-CONIC (ANALYTICAL)
             =2 -- VIRTUAL MASS (ANALYTICAL)
             =3 -- NUMERICAL DIFFERENCING USING VIRTUAL-MASS.
             --NOTE--IF THIS CODE IS NOT INPUT,ISTMC = 1 IS
             ASSUMED.
IDNF     --  DYNAMIC NOISE FLAG
             =0 -- DYNAMIC NOISE IS ZERO--ASSUMED IF NOT INPUT
             =1 -- DYNAMIC NOISE IS NOT ZERO
DNCN     --  CONSTANTS USED TO CALCULATE DYNAMIC NOISE (NEED BE
             INPUT ONLY IF IDNF=1)
IMNF     --  MEASUREMENT NOISE FLAG
             =0 -- MEASUREMENT NOISE IS CONSTANT (ASSUMED IF
                   NOT READ IN)
             =1 -- MEASUREMENT NOISE IS TO BE COMPUTED (THIS
                   OPTION IS NOT AVAILABLE WITH THIS PROGRAM).
MNCN     --  ARRAY OF VARIANCES FOR EACH TYPE OF MEASUREMENT
             --NOTE--IF THIS ARRAY IS OMITTED FROM THE NAMELIST,
             THE FOLLOWING ARRAY WILL BE ASSUMED.
             RANGE (IDEALIZED STATION)        1.E-6
             RANGE-RATE (IDEALIZED STATION)   1.E-12
             RANGE (STATION 1)                1.E-6
             RANGE-RATE (STATION 1)           1.E-12
             RANGE (STATION 2)                1.E-6
             RANGE-RATE (STATION 2)           1.E-12
             RANGE (STATION 3)                1.E-6
             RANGE-RATE (STATION 3)           1.E-12
             STAR ANGLE 1                     2.5E-9
             STAR ANGLE 2                     2.5E-9
             STAR ANGLE 3                     2.5E-9
             APPARENT PLANET DIAMETER         2.5E-9
NST      --  NUMBER OF TRACKING STATIONS ON THE ROTATING EARTH
             --NOTE--THIS INFORMATION NEED BE INPUT ONLY IF
             INCLUDED IN THE MEASUREMENTS IS TYPE 3,4,5,6,7,8
             IF NO INFORMATION ON THE TRACKING STATIONS IS INPUT
             THREE STATIONS WILL BE ASSUMED AS THE FOLLOWING
                                       ALT        LAT        LONG
             1.  GOLDSTONE     --  1.031 KM   35.384N   116.833W
             2.  MADRID        --   .050 KM   40.417N     3.667W
             3.  CANBERRA      --   .050 KM   35.311S   149.136E
SAL      --  ARRAY OF ALTITUDES OF EACH TRACKING STATION
SLAT     --  ARRAY OF LATITUDES OF EACH TRACKING STATION
SLON     --  ARRAY OF LONGITUDES OF EACH TRACKING STATION
             --NOTE--THE ABOVE THREE ARRAYS MUST BE INPUT ONLY
             IF NST.NE.0
U1,V1,W1--   DIRECTION COSINES OF STAR PLANET ANGLE 1
             (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
             IF THESE ARE NOT INPUT STAR NUMBER 1 IS ASSUMED
             TO BE CANOPUS WITH
             U1=-.061351,  V1=.237886,  W1=-.969355
```

```
U2,V2,W2--  DIRECTION CONSINES OF STAR PLANE ANGLE 2
            (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
            IF THESE ARE NOT INPUT STAR NUMBER 2 IS ASSUMED
            TO BE BETELGEUSE WITH
            U2=.028986,  V2=.960388,  W2=-.277141
U3,V3,W3--  DIRECTION COSINES OF STAR PLANET ANGLE 3
            (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
            IF THESE ARE NOT INPUT STAR NUMBER 3 IS ASSUMED
            TO BE RIGEL WITH
            U3=.201963,  V3=.831343,  W3=-.517784
FACP    --  POSITION FACTOR FOR NUMERICAL DIFFERENCING
            (NEED BE INPUT ONLY IF ISTMC=3)
            ASSUMED TO BE 1 KM IF NOT INPUT
FACV    --  VELOCITY FACTOR FOR NUMERICAL DIFFERENCING
            (NEED BE INPUT ONLY IF ISTMC=3)
            ASSUMED TO BE 1.E-4 KM/SEC IF NOT INPUT.
FOP     --  A VALUE TO BE USED AS AN OFF-DIAGONAL ANIHILATION
            ELEMENT IN THE EIGENVECTOR ROUTINE FOR POSITION
            EIGENVALUES AND EIGENVECTORS (ASSUMED TO BE 1.E-15
            IF NOT READ IN)
FOV     --  A VALUE TO BE USED AS AN OFF-DIAGONAL ANIHILATION
            ELEMENT IN THE EIGENVECTOR ROUTINE FOR VELOCITY
            EIGENVALUES AND EIGENVECTORS (ASSUMED TO BE 1.E-25
            IF NOT READ IN)
ISTM1   --  AN ALTERNATE STATE TRANSITION MATRIX CODE
            =0  IF DELTM IS GREATER THAN DTMAX (DESCRIBED
                BELOW) CALCULATE PSI BY USING THE SUN AS THE
                GOVERNING BODY.  (ASSUMED 0 IF NOT INPUT)
            =1  IF DELTM IS GREATER THAN DTMAX CALCULATE PSI BY
                NUMERICAL DIFFERENCING.
DTMAX   --  THE MAXIMUM DELTM (IN DAYS) SO THAT THE STATE
            TRANSITION MATRIX COMPUTATION IS CONSIDERED VALID
            WHEN USING EITHER THE PATCHED CONIC TECHNIQUE OR
            THE VIRTUAL MASS TECHNIQUE
            (ASSUMED TO BE 8 DAYS IF NOT READ IN)
NDACC   --  ACCURACY CODE FOR NUMERICAL DIFFERENCING
            =0  USE THE SAME ACCURACY FIGURE IN THE CALCULATION
                OF THE STATE TRANSITION MATRIX BY THE METHOD OF
                NUMERICAL DIFFERENCING AS IS USED IN THE
                NOMINAL TRAJECTORY (ASSUMED IF NOT INPUT)
            =1  CHANGE THE ACCURACY IN USING THE NUMERICAL
                DIFFERENCING METHOD TO ACCND (DESCRIBED BELOW)
ACCND   --  ACCURACY TO BE USED IN THE CALCULATION OF THE STATE
            TRANSITION MATRIX BY THE METHOD OF NUMERICAL
            DIFFERENCING.  (USED ONLY IF NDACC=1) ASSUMED TO BE
            2.5E-5 IF NOT INPUT.
DELAXS  --  SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL
            DIFFERENCING TO COMPUTE PSI AND H IF IAUG = 5, 9,
            OR 10.  (ASSUMED 100 KM IF NOT INPUT)
DELECC  --  ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING
            TO COMPUTE PSI AND H IF IAUG = 5, 9, OR 10.
            (ASSUMED 1.E-5 IF NOT INPUT)
DELICL  --  INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING
            TO COMPUTE PSI AND H IF IAUG = 5, 9, OR 10.
            (ASSUMED 10 ARCSECONDS IF NOT INPUT)
```

DELMUS -- FACTOR USED IN NUMERICAL DIFFERENCING FOR THE MU OF THE SUN TO GENERATE THE AUGMENTED STATE TRANSITION MATRIX WHEN IAUG = 3, 7, 9, OR 11. (ASSUMED 1.E7 WHEN NOT INPUT)

DELMUP -- FACTOR USED IN NUMERICAL DIFFERENCING FOR THE MU OF THE TARGET PLANET TO GENERATE THE AUGMENTED STATE TRANSITION MATRIX WHEN IAUG = 3, 7, 9, OR 11. (ASSUMED .1 WHEN NOT INPUT)

D. IN ORDER TO EXERCISE THE SIMULATION OPTION, A NAMELIST ENTITLED SMLTN IS READ WHICH CONTAINS ALL THE VARIABLES MENTIONED ABOVE FOR THE TRAJECTORY AND ERROR ANALYSIS MODES PLUS THE FOLLOWING.

NEV4 -- NUMBER OF QUASI-LINEAR FILTERING EVENTS TO BE RUN

T4 -- AN ARRAY OF TIMES AT WHICH QUASI-LINEAR FILTERING EVENTS ARE TO TAKE PLACE.
--NOTE--THIS ARRAY IS NECESSARY ONLY IF NEV4 IS NOT ZERO

ADEVX -- THE VECTOR DESCRIBING THE ACTUAL DEVIATION OF THE ACTUAL TRAJECTORY FROM THE MOST RECENT NOMINAL TRAJECTORY

BIA -- AN ARRAY OF MEASUREMENT BIASES WHICH DETERMINE THE ACTUAL VALUE TO BE USED FOR EACH OF THE TYPES OF MEASUREMENTS

DMUSB -- ACTUAL BIAS OF THE MU OF THE SUN TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
(ASSUMED TO BE ZERO IF NOT INPUT)

DMUPB -- ACTUAL BIAS OF THE MU OF THE TARGET PLANET TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
(ASSUMED TO BE ZERO IF NOT INPUT)

DAB -- ACTUAL BIAS IN THE SEMI-MAJOR AXIS OF THE TARGET PLANET TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
(ASSUMED TO BE ZERO IF NOT INPUT)

DEB -- ACTUAL BIAS IN THE ECCENTRICITY OF THE TARGET PLANET TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
(ASSUMED TO BE ZERO IF NOT INPUT)

DIB -- ACTUAL BIAS IN THE INCLINATION OF THE TARGET PLANET TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
(ASSUMED TO BE ZERO IF NOT INPUT)

TTIM1 -- THE FIRST TIME AT WHICH THE VALUES USED FOR THE ACTUAL UNMODELLED ACCELERATION WILL BE ALTERED

TTIM2 -- THE SECOND TIME AT WHICH THE VALUES USED FOR THE ACTUAL UNMODELLED ACCELERATION WILL BE ALTERED

UNMAC -- AN ARRAY OF VALUES WHICH DETERMINE THE ACTUAL UNMODELLED ACCELERATION TO BE USED AT A GIVEN TIME
NOTE--THESE VALUES ARE ASSUMED ZERO IF NOT INPUT

T0 - T1	T1 - T2	T2 - TF	
X1	X2	X3	ACCELERATION
Y1	Y2	Y3	ACCELERATION
Z1	Z2	Z3	ACCELERATION

```
SLB      --  AN ARRAY OF ACTUAL BIASES IN THE LOCATIONS OF THE
             THREE ROTATING STATIONS ON THE EARTH
             (AL1,LAT1,LONG1,AL2,LAT2,LONG2,AL3,LAT3,LONG3)
             NOTE--THESE VALUES ARE ASSUMED TO BE ZERO IF NOT
             INPUT
IAMNF    --  ACTUAL MEASUREMENT NOISE CODE
             =0  --  ASSUME THE ACTUAL UNCERTAINTIES IN THE
                     MEASUREMENT NOISE ARE THE SAME AS THE
                     UNCERTAINTIES ASSUMED IN THE MOST RECENT
                     NOMINAL TRAJECTORY
             =1  --  CALCULATE THE ACTUAL UNCERTAINITES IN THE
                     MEASUREMENT NOISE USING THE FOLLOWING
                     CONSTANTS
             NOTE  --  IF NOT INPUT IAMNF IS ASSUMED ZERO
AVARM    --  ACTUAL VARIANCES TO BE USED IN COMPUTING THE ACTUAL
             UNCERTAINTIES IN THE MEASUREMENT FROM WHICH THE
             ACTUAL MEASUREMENT NOISE IS CALCULATED
             --NOTE--NEED BE INPUT ONLY IF IAMNF=1
NBOD1    --  NUMBER OF BODIES TO BE CONSIDERED IN THE ACTUAL
             TRAJECTORY (ASSUMED TO BE 11 IF NOT INPUT)
NB1      --  AN ARRAY OF CODES OF PLANETS TO BE USED IN THE
             ACTUAL TRAJECTORY (IF NOT INPUT ALL MAJOR PLANETS
             IN THE SOLAR SYSTEM ARE CONSIDERED PLUS THE SUN AND
             THE EARTHS MOON)
ACC1     --  AN ACCURACY FIGURE TO BE USED IN THE VIRTUAL MASS
             PROGRAM WHEN GENERATING THE ACTUAL TRAJECTORY
             (IF NOT INPUT ACC1 IS ASSUMED TO BE 1.E-6)
ARES     --  AN ARRAY OF ACTUAL RESOLUTION ERRORS CORRESPONDING
             TO THE GUIDANCE EVENTS.  (ASSUMED 0 IF NOT INPUT)
             --NOTE--NEED BE INPUT ONLY IF GUIDANCE EVENTS OCCUR
APRO     --  AN ARRAY OF ACTUAL PROPORTIONALITY ERRORS FOR EACH
             GUIDANCE EVENT.  (ASSUMED ZERO IF NOT INPUT)
             --NOTE--NEED BE INPUT ONLY IF GUIDANCE EVENTS OCCUR
AALP     --  AN ARRAY OF ACTUAL ERRORS FOR POINTING ANGLE ONE
             FOR THE GUIDANCE EVENTS.  (ASSUMED ZERO IF NOT
             INPUT)  --NOTE--NEED BE INPUT ONLY IF GUIDANCE
             EVENTS OCCUR
ABET     --  AN ARRAY OF ACTUAL ERRORS FOR POINTING ANGLE TWO
             FOR THE GUIDANCE EVENTS.  (ASSUMED ZERO IF NOT
             INPUT)  --NOTE--NEED BE INPUT ONLY IF GUIDANCE
             EVENTS OCCUR
```

III. OUTPUT OPTIONS

All printed output that may be obtained from the various modes of operation of STEAP is described in this chapter.

A. Trajectory Mode

Initially, the input data are printed with the exercised options. The entries that appear in this group are listed below:

1) Initial trajectory time in both calendar date and Julian date;
2) Final trajectory time in both calendar date and Julian date;
3) Initial trajectory time in days;
4) Augmentation code (for the trajectory mode this code is always 1);
5) Initial state vector - the initial state is printed first in the coordinate system in which it is input and finally in heliocentric ecliptic coordinates;
6) Nominal trajectory module code;
7) Nominal trajectory information -

   Bodies to be considered,

   Target planet;
8) Length units per A. U.;
9) Time units per day;
10) If the ephemeris is to be computed at every time interval, a message to this effect will be printed. Otherwise, the orbital elements of the planets that will be used throughout the trajectory will be printed;
11) A message is printed that notifies the user if output is to be suppressed at initial and final steps in the virtual mass trajectory;
12) If the virtual mass program will integrate only until reaching the sphere of influence of the target planet, a message will be printed to that effect. However, if the trajectory will continue until reaching a normal stopping condition the user will be notified;

13) Accuracy figure;

14) Print intervals -

Days,

Increments.

Output may be obtained throughout the trajectory at specified times directly from the Virtual Mass Program. Three variables determine when this printout will be received: IPRINT, DELTP, and INPR.

If IPRINT = 0, the Virtual Mass Program will give printed output at both the initial trajectory time and the final trajectory time. Otherwise, no initial and final printout will be obtained. In the trajectory mode, IPRINT is assumed zero unless otherwise stated (see Chapter II, Input Options).

The variable DELTP specifies the number of days after which printed output will be received. DELTP is assumed 3 if not stated otherwise in the trajectory mode. In this situation, printout will be given after every three-day interval (see Input Options, Chapter II).

Printed output will be given after every INPR increment. For the trajectory mode INPR = 100 unless it is read in the NAMELIST as a different value (see Chapter II, Input Options).

If the printout is to occur after a given time increment as specified by one of the above three options, the following information is printed.

Block 1:

1) Trajectory time;

2) Total time increments used to date;

3) Spacecraft inertial trajectory

Block 2:

1) Calendar date;

2) Julian date;

3) Ephemeris data.

Block 3:

Spacecraft trajectories relative to planets.

Block 4:

1) Position;
2) Velocity;
3) Spacecraft position relative to virtual mass;
4) Spacecraft velocity relative to virtual mass;
5) Kepler vector;
6) Eccentricity vector;
7) Virtual mass magnitude;
8) Virtual mass magnitude rate.

Block 5:

Virtual mass positions relative to planets.

Finally, a summary of the trajectory mode is printed containing the following data:

1) Accuracy;
2) True anomaly increment;
3) Initial trajectory time together with its calendar date and Julian date;
4) Heliocentric ecliptic initial and final coordinates of the vehicle;
5) Position and velocity of the vehicle relative to the Earth at final time;
6) Position and velocity of the vehicle relative to the target planet at final time;
7) At closest approach the position and velocity of the vehicle relative to target planet;
8) If the vehicle encountered the sphere of influence of the target planet, the position and velocity of the vehicle relative to the target planet together with **B**, **B·T**, and **B·R** are printed. Otherwise, a message is printed stating the vehicle did not reach the sphere of influence of the target planet;
9) Total time increments used;
10) Total computer time used in the Virtual Mass Program.

B. Targeting Mode

The first page of output from the targeting mode includes preliminary data used in the program and is generally self-explanatory. The first section includes the input data as read into the computer. These data are described in Chapter II.

If the zero-iterate injection conditions are to be computed internally by the program, pertinent information related to the point-to-point conditions (on which the zero-iterate is based) is then summarized. These data are identical to portions of the output generated by the SPARC (ref 1) program and published in the numerous Earth-Planet Trajectories volumes. Those sections of the SPARC data not of immediate interest to the targeting program are omitted. The format for the printout of the point-to-point data is the same as that used by SPARC to allow easy comparison. The abbreviations used in listing this information is standard and will not be discussed here. For a detailed discussion of these data, see reference 1.

A summary of the injection conditions is then provided. The most important information here is the actual injection time along with the injection position and velocity (given in heliocentric ecliptic coordinates) when these are computed internally in the program. The injection position recorded here is never varied in the program so this is the actual initial position of the final (targeted) trajectory. The injection velocity is of course altered during the course of the targeting.

The target conditions are then summarized. If target options 3 or 4 are specified, the final time listed here is based on a hyperbolic trajectory from the sphere of influence to closest approach. The point-to-point injection conditions are computed using this corrected arrival date. If the target option is 5 or 6, the auxiliary sphere of influence conditions based on the point-to-point approach asymptote and the closest approach target conditions are recorded.

Finally, the target schedule is listed. This involves simply recording the progressive accuracy levels to be used in the targeting.

The next page of output deals with the progressive refinement of the injection velocity to obtain the targeted value. Here the output consists of two main types of data: data pertaining to each successive nominal and data related to the construction of each state transition matrix.

Upon integration of each nominal trajectory, a row of data is recorded. The first half of this row supplies the following information:

LEVEL ITER STEP ACCY XDOT YDOT ZDOT B·T B·R TSI

The LEVEL number corresponds to the current integration accuracy level. ITER lists the number of iterations made at the current level. STEP is always zero for the nominal trajectory. ACCY is the current integration accuracy. XDOT, YDOT, and ZDOT are the velocity components of the current iterate given in heliocentric ecliptic coordinates. The next data lists the values of the target variables realized on this iterate. In all target options these parameters initially consist of B·T, B·R, and t_{SI}. Here t_{SI} is given as a Julian date referenced to 1900. In the final targeting of option 6, the target parameters become i_{CA}, r_{CA}, and t_{CA}.

The latter half of the nominal trajectory data consists of the following:

TARG B·T TARG B·R TARG TSI INTEG TIME TOTAL TIME INTEG INCRS

The first three items are the desired target values. Again these are generally B·T, B·R, and t_{SI}. In target options 5 or 6 these target values, being a function of the actual approach asymptote, vary slightly with each iterate. In the final targeting of option 6 problems, the target parameters become i_{CA}, r_{CA}, t_{CA}. INTEG TIME is the computer time in seconds used in integrating that iterate. TOTAL TIME is the cumulative computer time in seconds used up to the printing of this row. INTEG INCRS refers to the total number of increments used in the integration of the current nominal.

In the construction of the state transition matrix three (or two in the case of target option 3) integrations are required. The data contained in the first half of the nominal trajectory row are recorded for each of these integrations. However, instead of the latter half of that row, the actual state transition matrix based on the perturbed trajectories is recorded.

Whenever outer targeting is required, a statement is printed giving the radius and date of closest approach to the target planet along with the computed radius of the artificial sphere of influence. When a bad-step correction is used, a message to that effect is written.

The targeting procedure ends when an acceptable injection velocity is determined or when the maximum number of iterations has been made. At this point a targeting summary is given. The injection position and velocity are output both in heliocentric ecliptic and geocentric ecliptic coordinates. The injection time is given to thousandths of seconds. The actual trajectory target values corresponding to these injection conditions are then listed along with the desired target conditions.

The final output from the targeting program results from operating the trajectory mode in its normal state. The appropriate flags described in Chapter III. A are set so that information from this integration is printed at the initial time, at intermediate points determined by INCPR and TIMPR (see Chapter II.B, Input Option Targeting Mode), at the sphere of influence, and at the first time determined as the point of closest approach to the target planet.

C. Error Analysis Mode

As in the trajectory mode, the input data are printed initially. All those items listed at the beginning of the trajectory mode are printed for the error analysis mode. See Input Options, Chapter II, for differences in assumed values in this mode of operation.

In addition, the following items are listed:

1) Measurement schedule;
2) Event schedule;
3) Initial covariance matrix;
4) State transition matrix code. If the state transition matrix is to be generated using numerical differencing, the position and velocity factors are printed;
5) Dynamic noise constants;
6) Measurement noise constants;
7) Station location constants.

The same options (IPRINT, DELTP, and INPR) are available for printout from the Virtual Mass Program as described for the trajectory mode. If IPRINT is not otherwise stated it is assumed equal to one in the error analysis mode. Thus if DELTP (assumed 1.E+50 days) and INPR (assumed 7777777) are large enough, no printout from the Virtual Mass Program itself will be obtained. However, if printout is needed, it will occur as the three variables mentioned above specify and the same information will be printed as stated for the trajectory mode.

When a measurement is processed, the following information will be printed:

1) Initial trajectory time in days together with calendar date and Julian date;
2) Final trajectory time in days together with calendar date and Julian date;
3) State vector at initial and final time;
4) Position and velocity of vehicle relative to Earth and target planet at both initial and final time;
5) Number of measurement;
6) Type of measurement;
7) State transition matrix;
8) Diagonal of dynamic noise matrix;
9) Observation matrix;
10) Measurement noise matrix;
11) **Gain Matrix** - K;
12) Covariance matrix just before the measurement;
13) Covariance matrix after the considering the measurement.

During an eigenvector event the following data are printed:

1) State vector at the time of the eigenvector event;
2) State transition matrix;
3) Diagonal of dynamic noise matrix;
4) Covariance matrix at time of eigenvector event;

5) Position and velocity eigenvalues and eigenvectors as specified by IEIG (see Chapter II);

6) Hyperellipsoids for both position and velocity portions of the covariance matrix for sigma levels as specified by IHYP1 (see Chapter II);

7) Correlation coefficient matrix at time of eigenvector event.

For a prediction event, the printed output will include the following items:

1) State vector at time of prediction event;

2) State transition matrix relating the deviations at the time of the prediction event to those at the time of the last measurement or event;

3) Diagonal of dynamic noise matrix;

4) Covariance matrix at time of prediction event;

5) State transition matrix relating deviations at the prediction time to those at the time of the prediction event;

6) Diagonal of dynamic noise matrix;

7) Covariance matrix at prediction time;

8) Position and velocity eigenvalues and eigenvectors as specified by IEIG;

9) Hyperellipsoids for both the position and velocity portions of the covariance matrix at prediction time for sigma levels according to IHYP1;

10) Correlation coefficient matrix at prediction time;

11) If the prediction time is within one day of when the vehicle reaches the sphere of influence of the target planet, the covariance matrix of uncertainties in B•T and B•R is printed together with its eigenvectors and eigenvalues and the associated hyperellipsoids.

For a guidance event the following data are printed:

1) State vector at time of guidance event;

2) State transition matrix relating deviations at the time of the guidance event to those at the time of the last measurement or event;

3) Diagonal of dynamic noise matrix;

4) Covariance matrix at time of guidance event;

5) Position and velocity eigenvalues and eigenvectors and related hyperellipsoids according to IEIG and IHYP1;

6) State transition matrix relating deviations at the time of the guidance event to those at the time of the last guidance event;

7) Diagonal of dynamic noise matrix;

8) Covariance matrix relating the time of this guidance event to that at the time of the last guidance event;

9) Position and velocity eigenvalues and eigenvectors and associated hyperellipsoids.

At this point a decision is made as to which guidance policy is being used. If the fixed time-of-arrival policy is used the following data are printed:

1) Time at which vehicle encounters closest approach of target planet;

2) Position and velocity of vehicle at closest approach;

3) State transition matrix relating deviations at the time of closest approach to those at the time of the guidance event;

4) Variation matrix;

5) Uncertainty in target conditions before correction together with its eigenvalues and eigenvectors and associated hyperellipsoids;

6) Guidance matrix.

From this point printout is again independent of the guidance policy.

If the two-variable B-plane policy is used the following data are printed:

1) Time at which vehicle reaches sphere of influence;

2) Position and velocity of vehicle at sphere of influence together with B, B•T, and B•R;

3) State transition matrix relating deviations at sphere of influence to those at the time of the guidance event;

4) Partial of B·T and B·R with respect to the state vector;

5) Guidance submatrices A and B;

6) Uncertainty in target conditions before correction with its eigenvalues and eigenvectors and associated hyper-ellipsoids;

7) Guidance matrix.

From this time printout for the three policies is identical.

If the three-variable B-plane policy is used the following data are printed:

1) Time at which vehicle enters sphere of influence of target planet;

2) Position and velocity of vehicle at sphere of influence in addition to B, B·T, and B·R;

3) Variation matrix;

4) Uncertainty in target conditions before correction together with eigenvalues and eigenvectors and hyper-ellipsoids;

5) Guidance matrix.

The rest of the printout is independent of the type of guidance policy being used, and is as follows:

1) Covariance matrix associated with velocity components;

2) Expected value of ΔV;

3) Standard deviation of expected value of ΔV;

4) If the execution error code is 1 (see Input Options), the eigenvalues and eigenvectors of the above covariance matrix and the expected value of the velocity correction are printed;

5) Execution error matrix in addition to its eigenvalues and eigenvectors and hyperellipsoids;

6) Modified covariance matrix at the time of the guidance event;

7) Uncertainty in target conditions after correction together with its eigenvalues and eigenvectors and hyperellipsoids.

In addition to the output categorized above, if the vehicle encounters sphere of influence or closest approach during the course of the nominal trajectory, the pertinent information related to the encounter is printed.

A summary of the error analysis mode is printed containing the following items:

1) Accuracy figure;

2) True anomaly increment;

3) Length units per A. U.;

4) Time units per day;

5) A message is printed stating in what manner the orbital elements of the planets are calculated;

6) Initial trajectory time together with its calendar date and Julian date;

7) Final trajectory time and its calendar date and Julian date;

8) Heliocentric ecliptic coordinates of the vehicle at both initial and final times;

9) Position and velocity of the vehicle relative to the Earth and the target planet at final time;

10) The time at closest approach in addition to the position and velocity of the vehicle relative to the target planet at closest approach;

11) If the vehicle did not reach the sphere of influence of the target planet a message to that effect is printed. Otherwise, the time at which it entered the sphere of influence is printed together with the position and velocity of the vehicle relative to the target planet and B, B·T, and B·R;

12) Total time increments;

13) The method by which the state transition matrix is computed together with any limitations;

14) Number of measurements taken;

15) Number of events having occurred plus the number of each type of event;

16) For guidance events, the variances used for resolution error, proportionality error, error in pointing angle 1, and pointing angle 2;

17) Dynamic noise constants;

18) Measurement noise constants;

19) Direction cosines for three star planet angles;

20) State vector at initial and final times;

21) Initial covariance matrix;

22) Final covariance matrix.

D. Simulation Mode

The input data are printed initially. The same input items are printed as stated for the error analysis mode. In addition, the following items are printed:

1) Actual deviation of state vector at initial time;

2) Bodies to be considered in actual trajectory;

3) Accuracy figure for actual trajectory;

4) Actual measurement biases;

5) Dynamic constant biases to be used in actual trajectory;

6) Actual unmodelled acceleration to be used to calculate the actual dynamic noise;

7) Biases in station location constants;

8) Actual measurement noise constants. If these are to be the same as the estimated measurement noise constants, a message to this effect is printed.

The same comments concerning the printout of the Virtual Mass Program are applicable to the simulation mode as in the error analysis mode.

The following measurement cycle information will be printed when a measurement is processed:

1) Initial trajectory time;
2) Final trajectory time;
3) State vector at initial time of original nominal, most recent nominal, and actual trajectory;
4) State vector at final time of original nominal, most recent nominal, and actual trajectory;
5) Position and velocity of vehicle relative to Earth and target planet at initial and final time on original nominal, most recent nominal, and actual trajectory;
6) Number of measurement;
7) Type of measurement;
8) State transition matrix;
9) Diagonal of dynamic noise matrix;
10) Observation matrix;
11) Measurement noise matrix;
12) Gain Matrix - K;
13) Covariance matrix before the measurement;
14) Covariance matrix after the measurement;
15) Actual dynamic noise;
16) Matrix of variance of actual measurement noise;
17) Actual measurement noise;
18) Estimated and actual measurement;
19) Residual;
20) Residual uncertainties;
21) Estimated and actual deviation of the state vector from the most recent nominal;
22) Estimated and actual deviation of the state vector from the original nominal;
23) Actual orbit determination inaccuracy.

When an eigenvector event occurs, the following printout is obtained:

1) State vector at time of eigenvector event of original nominal, most recent nominal, and actual trajectory;
2) State transition matrix;
3) Diagonal of dynamic noise matrix;
4) Covariance matrix at time of eigenvector event;
5) Position and velocity eigenvalues and eigenvectors according to the specifications of IEIG, together with the related hyperellipsoids as IHYP1 specifies (see Input Options);
6) Correlation coefficient matrix at time of eigenvector event;
7) Actual dynamic noise;
8) Estimated and actual deviation of the state vector from the most recent nominal.

When a prediction event occurs, the printed output will include the following:

1) State vector at time of prediction event on original nominal, most recent nominal, and actual trajectory;
2) State transition matrix relating deviations at the time of the prediction event to those at the time of the last measurement or event;
3) Diagonal of dynamic noise matrix;
4) Covariance matrix at time of prediction event together with position and velocity eigenvalues and related hyperellipsoids;
5) Correlation coefficient matrix;
6) Actual dynamic noise;
7) Estimated and actual deviation of the state vector from the most recent nominal;
8) State transition matrix relating deviations at the prediction time to those at the time of the prediction event;
9) Diagonal of dynamic noise matrix;

10) Covariance matrix at prediction time together with eigenvalues and eigenvectors and related hyperellipsoids;

11) Correlation coefficient matrix;

12) If the prediction time is within one day of the time at which the vehicle reaches the sphere of influence on the original nominal trajectory, the covariance of uncertainties in B·T and B·R is printed together with its eigenvalues, eigenvectors, and hyperellipsoids.

At a guidance event the following items are included in the printout:

1) State vector at time of guidance event on original nominal, most recent nominal, and actual trajectory;

2) State transition matrix relating deviations at the time of the guidance event to those at the time of the last measurement or event;

3) Diagonal of dynamic noise matrix;

4) Covariance matrix at the time of the guidance event together with its eigenvalues and eigenvectors and related hyperellipsoids;

5) Actual dynamic noise;

6) Estimated and actual deviation of the state vector from the most recent nominal;

7) State transition matrix relating deviations at the time of the guidance event to those at the time of the last guidance event;

8) Diagonal of dynamic noise matrix;

9) Covariance matrix relating the time of this guidance event to the time of the last guidance event in addition to its eigenvalues, eigenvectors, and hyperellipsoids.

The next portion of printout from the guidance event depends on the type of guidance policy used. If the fixed-time-of-arrival policy is used the following data are printed:

1) Time at which the vehicle reached closest approach on the original nominal trajectory together with its position and velocity relative to the target planet. Partial of $B \cdot T$ and $B \cdot R$ with respect to the state vector (M-matrix);

2) Time at which the vehicle reached closest approach on the most recent nominal together with its position and velocity relative to the target planet;

3) State transition matrix relating deviations at the time at which the vehicle reached closest approach on the most recent nominal to those at the time of the guidance event;

4) Variation matrix;

5) Uncertainty in target conditions before correction together with its eigenvalues, eigenvectors, and hyperellipsoids;

6) Guidance matrix.

The rest of the printout is identical with that of the other guidance policies.

If the two-variable B-plane policy is used the following data are printed:

1) The time at which the vehicle reached the sphere of influence of the target planet on the original nominal plus its position and velocity relative to the target planet and B, $B \cdot T$, and $B \cdot R$;

2) Partials of $B \cdot T$ and $B \cdot R$ with respect to the state vector (M-matrix);

3) Time at which the vehicle reached the sphere of influence on the most recent nominal trajectory plus its position and velocity relative to the target planet and B, $B \cdot T$, and $B \cdot R$;

4) State transition matrix relating deviations at the time when the vehicle entered sphere of influence to those at the time of the guidance event;

5) Partials of $B \cdot T$ and $B \cdot R$ with respect to the state vector of the most recent nominal;

6) Guidance submatrices A and B;

7) Uncertainty in target conditions before correction together with its eigenvalues, eigenvectors, and hyperellipsoids;

8) Guidance matrix.

The printout which follows this is independent of the type of guidance policy.

If the three-variable B-plane policy is used the following data are printed:

1) Time at which the vehicle entered the sphere of influence of the target planet on the original nominal plus its position and velocity relative to the target planet and B, B•T, and B•R;

2) Partial of B•T and B•R with respect to the state vector (M-matrix);

3) Time at which the vehicle entered the sphere of influence on the most recent nominal trajectory plus its position and velocity relative to the target planet and B, B•T, and B•R;

4) Variation matrix;

5) Uncertainty in target conditions before correction together with its eigenvalues, eigenvectors, and hyperellipsoids;

6) Guidance matrix.

The additional printout is independent of the type of guidance policy except where noted:

1) Covariance matrix associated with velocity components plus its eigenvalues, eigenvectors, and hyperellipsoids (hyperellipsoids are not printed in the case of two variable B-plane guidance due to the singularity of the matrix);

2) Estimated and actual deviation of the state vector from the original nominal;

3) Commanded correction;

4) Perfect correction;

5) Commanded ΔV;

6) Error in correction due to navigation uncertainty;

7) Execution error matrix plus its eigenvalues, eigenvectors, and hyperellipsoids;

8) Modified covariance matrix at time of guidance event together with its eigenvalues, eigenvectors, and hyperellipsoids;

9) Uncertainty in target conditions after correction plus its eigenvalues, eigenvectors, and hyperellipsoids;

10) Actual error in correction;

11) Actual correction;

12) Actual error at target after correction.

For a quasi-linear filtering event the following output will be obtained:

1) State vector at time of quasi-linear filtering event on original nominal, most recent nominal, and actual trajectory;

2) State transition matrix;

3) Diagonal of dynamic noise matrix;

4) Covariance matrix at time of quasi-linear filtering event;

5) Correlation coefficient matrix;

6) Actual dynamic noise;

7) Estimated and actual deviations of state vector from most recent nominal;

8) State vector of "new" nominal at time of quasi-linear filtering event;

9) New actual deviation of state vector from most recent nominal.

Upon encountering the sphere of influence or closest approach of the target planet on any of the three trajectories, the pertinent information is printed.

A summary of the simulation mode includes the following items:

1) Accuracies used in both nominal and actual trajectories;

2) Bodies considered in both nominal and actual trajectories;

3) Gravitational constant biases used in actual trajectory;

4) Ephemeris biases used in actual trajectory;

5) Initial trajectory time;

6) Final trajectory time;

7) At initial time, position, and velocity of vehicle relative to Sun, Earth, and target planet;

8) At final time, position, and velocity of vehicle relative to Sun, Earth, and target planet on original nominal, most recent nominal, and actual trajectory;

9) Time at closest approach plus position and velocity of vehicle relative to target planet on all three trajectories;

10) The time at which the vehicle enters the sphere of influence of the target planet in addition to the position and velocity of the vehicle relative to the target planet and B, B•T, and B•R on all three trajectories;

11) Method by which the state transition matrix is computed in addition to its limitations;

12) Number of measurements taken;

13) Number of events plus the number of each type of event;

14) Variances of errors used in guidance events;

15) Actual errors used in guidance events;

16) Station location constants;

17) Dynamic noise constants;

18) Actual unmodeled acceleration;

19) Estimated measurement noise constants;

20) Actual measurement noise constants;

21) Direction cosines for three star planet angles;

22) Initial state vector for both nominal and actual trajectories;

23) Final state vector for all three trajectories;

24) Deviation of state vector from most recent nominal at final time;

25) Deviation of state vector from original nominal at final time;

26) Actual orbit determination inaccuracy at final time;

27) Initial covariance matrix;

28) Final covariance matrix.

IV. MAIN PROGRAM STRUCTURE

The main program is the routine that mechanizes the complete program for establishing the mode of operation and controls the computational process in an orderly, efficient manner. To accomplish this task and for ease in development and checkout, the program has been constructed in a series of major modules, each of which is itself divided into a number of subroutines. These modules serve to read the input data, generate a nominal trajectory, determine the measurement schedules, check the various tracking stations, sequence the specific events to be used, and finally process the data. In summary, each module performs a specified logical task. In its simplest form, the main program serves as a link between these modules.

The main program logic for using the trajectory mode, targeting mode, error analysis mode, and simulation mode can best be exemplified by the simplified flow chart shown in figure 1. A complete detailed flow chart is presented in Chapter V. The targeting mode is run as a separate program and is shown as a dotted line in figure 1. To run the error analysis or simulation mode a set of injection conditions are necessary. Since these initial conditions are established by the targeting mode (unless specified from another source), and are an input to the other three modes of the program, the targeting mode is supplied as a separate program. This allows the user to look over the results of a targeting run before using the other three modes of STEAP.

To completely understand the logic in each of the operational modes, the four sections of this chapter review the computational logic for each mode. In addition, each subroutine used in the program is discussed in Chapter V.

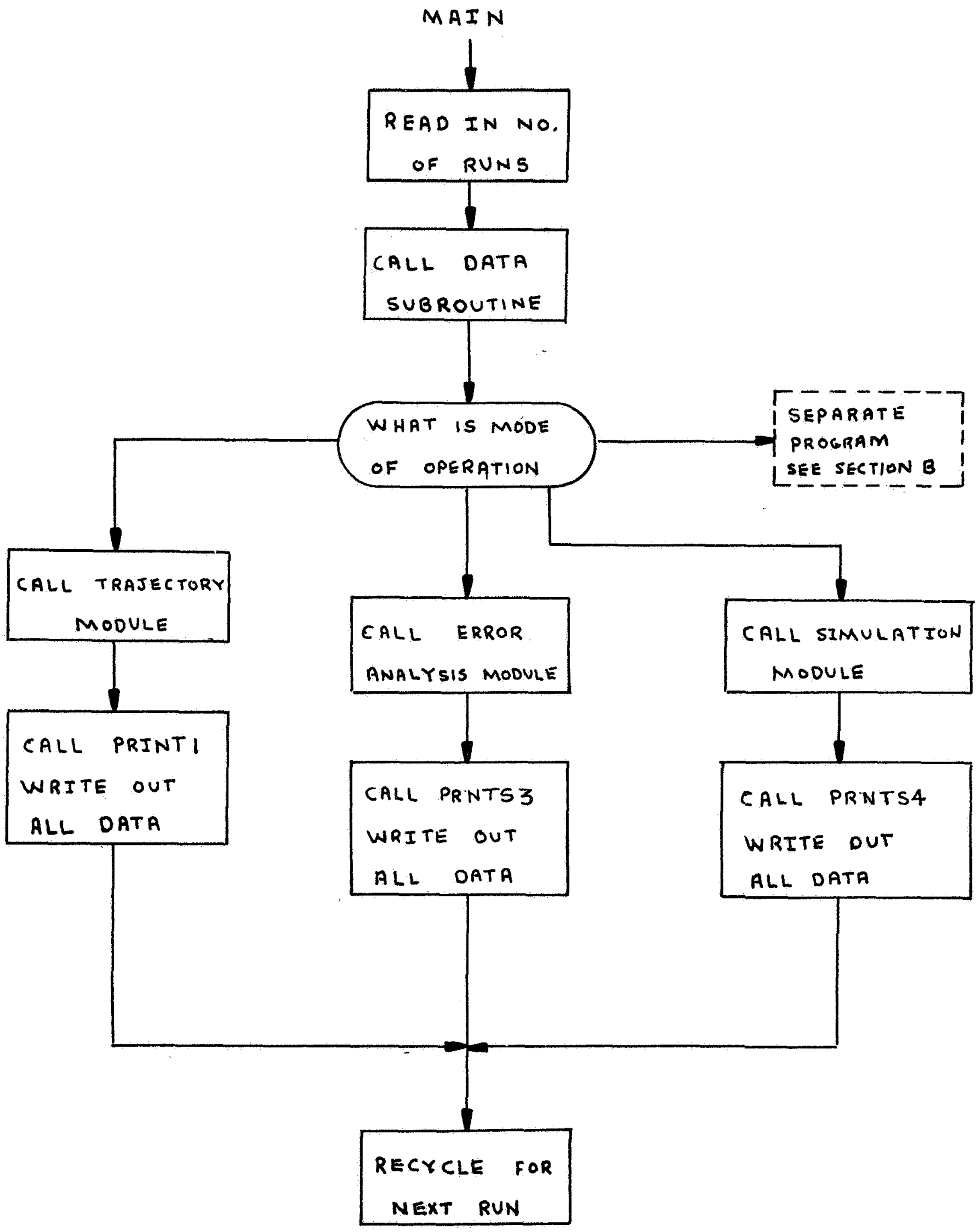


Figure 1.- Simplified Schematic of Main Program

A. Trajectory Mode Logic

Referring to figure 1, the program starts by reading the first data card to determine the number of runs to be made and then calls DATA. DATA supplies all the necessary input for each of the four operational modes. The first card in DATA specifies the mode to be exercised. In this section the trajectory mode sequence will be discussed. On returning from DATA with a set of initial conditions, the NTM module is called. This module determines the method by which the nominal trajectory will be calculated. The NTM module in turn calls the virtual mass subroutine VMP to generate the trajectory. Returning to the main program, a check is made to determine if the final trajectory time (specified by DATA) has been reached. If it has not, the NTM module is called again for the next increment or time interval. This process is repeated until the final trajectory time is reached, at which time the PRINT1 routine is called. PRINT1 is responsible for printing out all the virtual mass trajectory data. The program then recycles for the next run if specified by input, otherwise the program is terminated.

B. Targeting Mode Logic

Before discussing the actual program logic the general purpose and scope of the targeting mode will be reviewed. The purpose of the targeting mode is to generate a set of injection conditions which, when integrated forward in a trajectory model (integration accuracy level, gravitational bodies considered, etc) specified by the user, yield a trajectory satisfying prescribed mission requirements. The general mission conditions include a launch date, target date, launch planet, and target planet. More specific constraints are imposed as target conditions near the destination planet.

Six options are allowed in the specification of these target conditions. The first two options are really auxiliary to the remainder. In these options a heliocentric arc determined by the general mission conditions is patched to an earth-centered hyperbola consistent with a typical launch profile originating from Cape Kennedy on the initial date. The injection conditions are then computed from this crude trajectory. In the first option the conditions, termed the point-to-point conditions, are corrupted by a bias that improves their validity in forming an initial iterate in targeting n-body trajectories. In the second

option (the patched conic conditions) the bias is not included so that the generated set of injection conditions is a good initial iterate in obtaining targeted patched conic trajectories.

In the remaining options the injection conditions are generated that yield n-body trajectories consistent with more specific target constraints. In the third option these target conditions include the impact plane parameters B·T and B·R. The time at intersection of the sphere of influence t_{SI} is only approximated in this option. In the fourth option the impact plane parameters B·T and B·R along with the time t_{SI} are used as target constraints. The fifth and sixth targeting options are both based on the radius at closest approach r_{CA}, the inclination (with respect to the target planet equatorial plane) at closest approach i_{CA}, and the time at closest approach t_{CA}. These closest approach conditions may be converted to sphere of influence conditions B·T, B·R, and t_{SI}. In the fifth option injection conditions are generated consistent with these approximating sphere of influence conditions. In the last option, the option 5 injection conditions are first computed and these values are then refined to satisfy the exact closest approach conditions to the desired tolerances. These target options are summarized in table 1.

TABLE 1.- SUMMARY OF TARGET OPTIONS

Option	Title	Required input
1	Point-to-point	t_i, t_T, m_i, m_T
2	Patched conic	t_i, t_T, m_i, m_T
3	Two-variable SOI	t_i, t_T, m_1, m_2, ..., m_n, ACC,
		B·T, B·R, T_{SI}, ΔB·T, ΔB·R,
4	Three-variable SOI	t_i, t_T, m_1, m_2, ..., m_n, ACC,
		B·T, B·R, t_{SI}, ΔB·T, ΔB·R, Δt_{SI}
5	Approximate CA	t_i, t_T, m_1, m_2, ..., m_n, ACC,
		i_{CA}, r_{CA}, t_{CA}, ΔB·T, ΔB·R, Δt_{SI}
6	Strict CA	t_i, t_T, m_1, m_2, ..., m_n, ACC,
		i_{CA}, r_{CA}, t_{CA}, Δi_{CA}, Δr_{CA}, Δt_{CA}

Two options are provided in determining injection conditions. The injection conditions may either be read in as input data or computed internally in the point-to-point option. The first option provides the capability for handling problems such as mid-course correction analyses or multiplanet swingby targeting as well as permitting the efficient completion of partially targeted problems.

A final option is allowed in the specification of the integration accuracy level scheduling. For efficiency the preliminary (targeting) and state transition matrix computations are done at the first accuracy level. The final state transition matrix computed at the first level is then used repeatedly at the higher accuarcy levels to obtain improved velocity iterates. Lowering the first accuracy level results in more efficient preliminary targeting and state transition matrix computation. However the greater the difference between the first and last accuracy levels, the more likely it is that the original state transition matrix will lose validity at the final accuracy level. Permitting the user to choose the specific accuracy schedule therefore allows him to be as efficient or secure as he desires to be.

To understand the structure of the targeting mode, it is helpful to refer to the schematic diagram of the main program shown in figure 2. A detailed flow chart of the targeting main program is provided in Chapter V.

The program begins by reading the data for the specific problem under investigation. A discussion of the input data may be found in Chapter II of this volume.

If the zero iterate injection conditions are not read in as data (INJEK = 2), a set is computed internally in the program (INJEK = 1). The complex subroutine NJEXN is then responsible for this task. NJEXN computes these injection conditions by patching a heliocentric arc determined by the broad mission constraints to an Earth-centered hyperbola consistent with a standard launch profile from Cape Kennedy.

The program then enters the basic numerical differencing cycle. It first targets to sphere of influence conditions, regardless of the targeting option. If either of the closest approach options (options 5 or 6) are being used, auxiliary sphere of influence targets are computed at the completion of each nominal trajectory integration by calling the subroutine CASOI. All integrations in the cycle use the VMP subroutine and its associated subroutines (VECTOR, VMASS, ORB, EPHEM, etc).

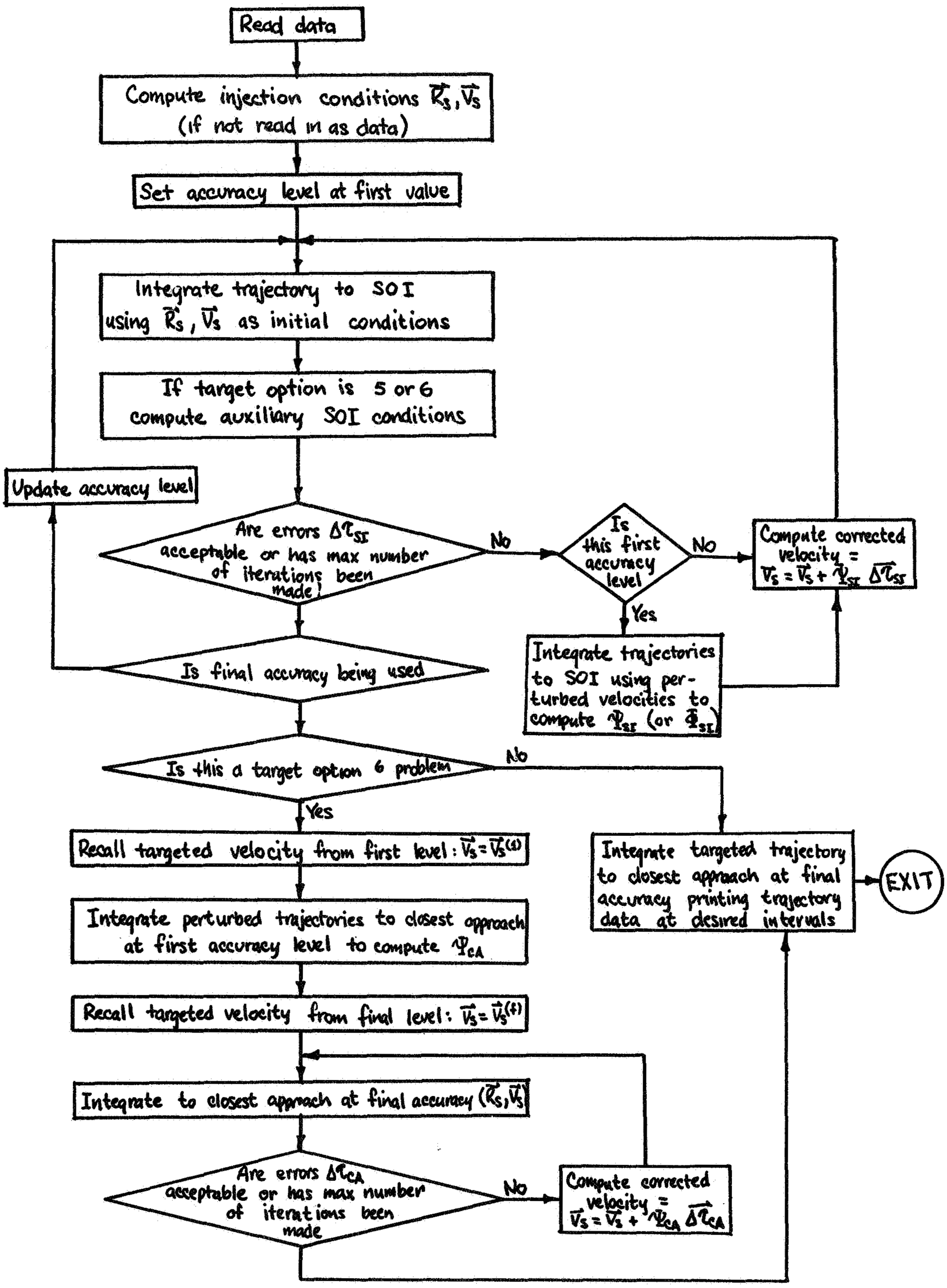


Figure 2.- Schematic Diagram of Targeting Program

The targeting is done in progressive "levels" corresponding to increasing integration accuracies. At the n^{th} level, the current iterate injection conditions are integrated to the sphere of influence. The target parameters are then evaluated. If these parameters are not within specified tolerances of their target values and if the number of iterations has not exceeded its maximum allowable figure for that level, a new iterate injection velocity is computed. It is computed by using a state transition matrix that relates changes in the target conditions (at the sphere of influence) to changes in the injection velocity components. If the accuracy level is at its lowest value, this state transition matrix is computed numerically about each nominal; at higher accuracy levels the final state transition matrix calculated at the lowest accuracy level is used repeatedly. Eventually the target errors will be acceptable or the maximum number of iterations will be made. At this point the accuracy will be increased to the next level and the entire process repeated.

When the targeting has been accomplished at the final accuracy level, the program checks the targeting option. If the targeting option is any of 3, 4, or 5, the program integrates the targeted velocity to the closest approach of the target parameter at the final accuracy level, recording the trajectory data at specified intervals. It then returns to the start of the program to accept the next problem.

If the targeting option is option 6, more work must be done. The final (targeted) velocity generated at the first accuracy level is recalled. It is integrated at the first level to closest approach of the target planet and the target parameters recorded. Three integrations are made using perturbed velocity components at this first integration accuracy to construct a state transition matrix now relating changes in the closest approach conditions to changes in the injection velocity components. The program now returns to the final accuracy level. It recalls the injection velocity targeted to the auxiliary sphere of influence conditions at this level and now integrates it to closest approach. If the closest approach errors are unacceptable, it uses the closest approach state transition matrix just computed at the low accuracy to predict an improved iterate. The process is repeated until either acceptable errors are encountered or the maximum allowable iterations have been made. The program then makes a final integration to closest approach, recording the trajectory at the desired intervals, before returning to the next problem.

C. Error Analysis Mode Logic

Returning from the DATA subroutine as depicted in figure 1, the error analysis mode logic starts out by calling the SCHED subroutine. The basic flow of the error analysis made is shown in figure 3. Following figure 3, SCHED sets up the measurement schedule by properly sequencing the times of observations as specified at input. It returns control to the main program with the type of measurement (range, range rate, or onboard types) and the time the next measurement is to be made. The time interval DELTM is then computed. The program next encounters a logical IF statement to determine if the latest trajectory time TRTM2 has gone past an event time. The events that can be encountered are eigenvector, prediction, and guidance. The times of these events are specified by input. If there is no event scheduled between two measurements, the NTM module is called with the set of initial conditions. The NTM module calls on the virtual mass subroutine VMP and returns with the final trajectory conditions at the end of the time interval DELTM. Returning to the main program, the state transition matrix module PSIM is called with the time interval DELTM, initial state vector, and code for the method to compute the state transition matrix. The PSIM module then calls one of these subroutines, depending on the code ISTMC which designates the computational procedure used for the state transition matrices. The state transition matrix is computed by NDTM (numerical differencing), or PCTM (patched conic)or virtual mass . Once the matrix is computed, control is returned to the main program where the next subroutine DYNØ is called. DYNØ then computes the dynamic noise matrix and returns. The next subroutine called is TRAKM which is responsible for generating the observation matrix. TRAKM is called with the final trajectory conditions at the end of the time interval and with a code that specifies what type of tracking will be used. On returning with the observation matrix the MENØ subroutine is called. MENØ is used to compute the measurement noise. When control is returned to the main program, the final subroutine that is called is NAVM. The navigation module NAVM contains all the necessary estimation and filtering equations to compute covariance matrices and gain matrices. After the necessary computations are made, NAVM returns control to the main program. PRINT3 is then called to write out all the necessary data. The entire process is then continually repeated until the final trajectory time is reached, at which time the problem is terminated.

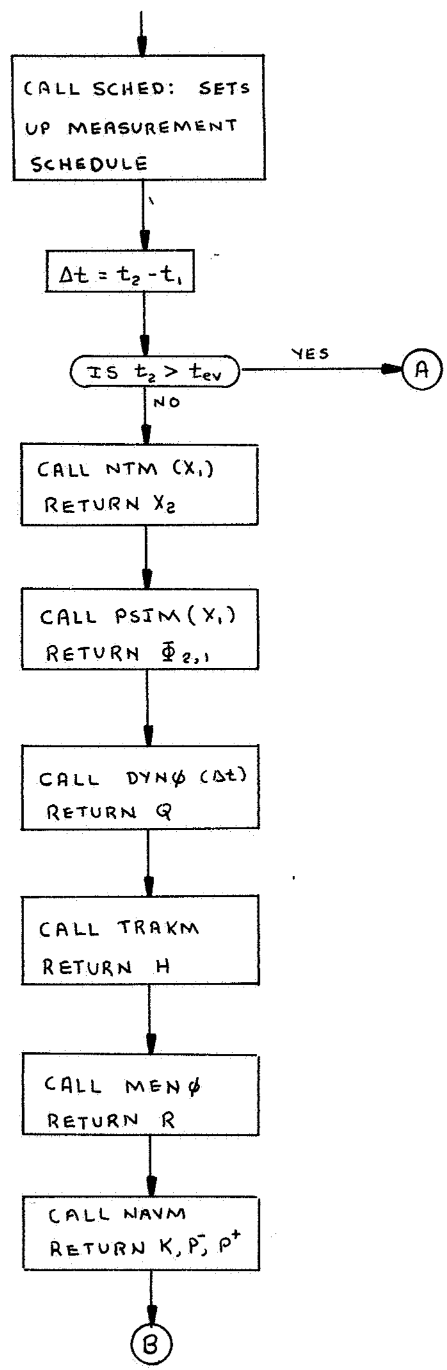


Figure 3.- Error Analysis Mode Logic

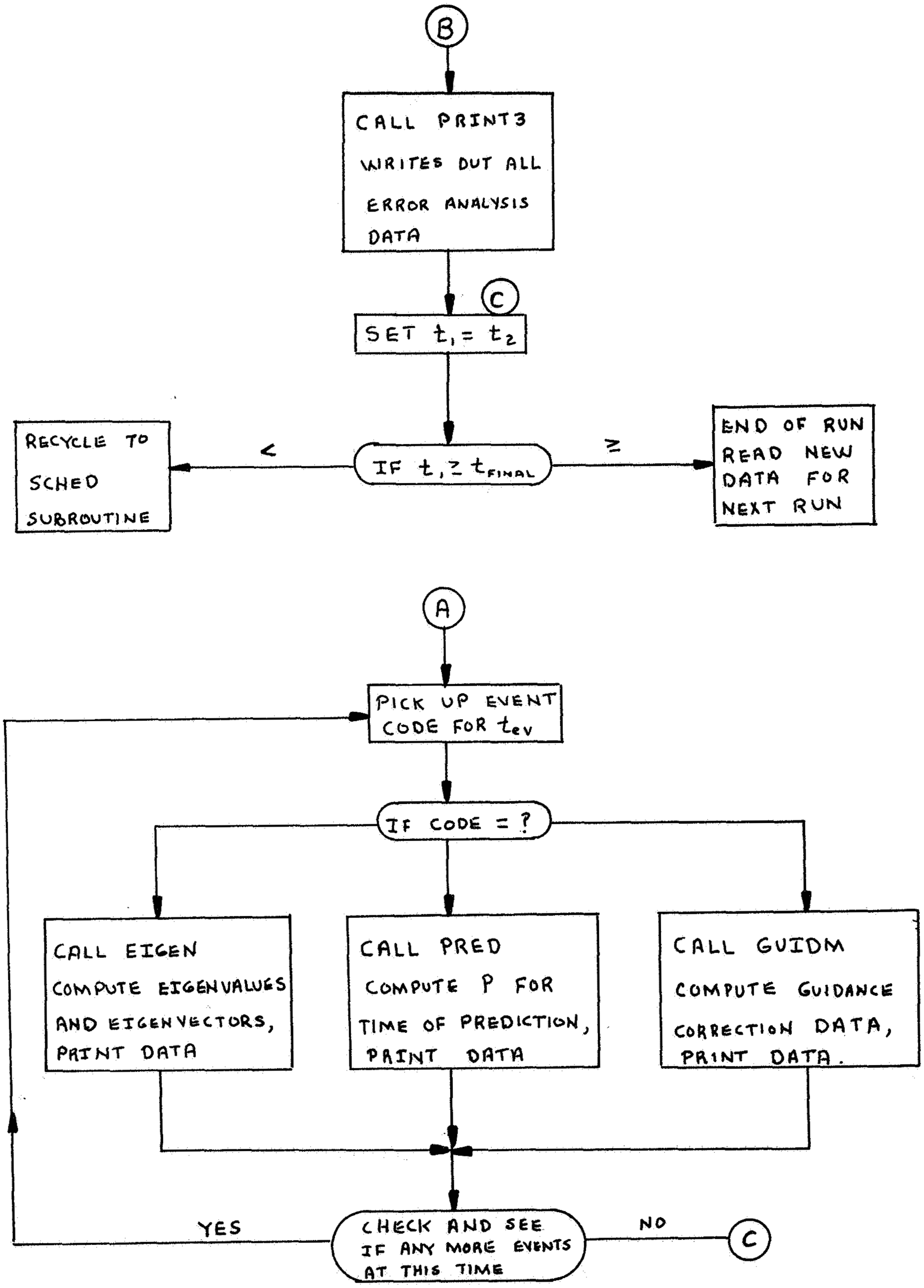


Figure 3.- Concluded

To complete the discussion of the error analysis logic, the logical process followed when an event time is encountered should be presented. The event decision is made at the logical IF statement at the beginning of the program. If an event time is encountered, one of three subroutines is called depending on the code for the type of event, PRED for a prediction event, GUID for a guidance event, and EIGEN for an eigenvector event. The EIGEN routine computes the necessary eigenvectors and eigenvalues. PRED determines the covariance matrix at some future critical time under the assumption that no further measurements are made. The guidance subroutine GUID determines by code what guidance law will be used (three are possible) and computes the required guidance matrix. Control is returned to the main program after any event computations at which time the time increment DELTM is updated and the process repeated just at when no events have occurred.

When the final trajectory time is reached PSIM, DYNØ, and NAVM are called to update the state transition matrix, and covariance matrix, PRNTS3 is then called to print out the final data. If there are no additional problems to be run, the program is terminated.

D. Simulation Mode Logic

The logic for the simulation mode starts after the DATA subroutine is called. A simplified schematic of the basic cycle of the simulation mode is presented in figure 4. The first subroutine called is SCHED. This routine is also used in the error analysis mode. After returning from SCHED the next time interval DELTM is computed and the NTM module is called with the initial trajectory conditions of the original nominal trajectory. The NTM module calls the virtual mass trajectory subroutine VMP and computes the trajectory conditions of the original nominal at the end of the time interval. The NTM module used at this point in the program, which is discussed in more detail in the analytic manual, uses assumed dynamics to generate the original nominal trajectory. The next step in the process is an IF statement that determines if a quasilinear filtering event has taken place. For now, assume that no quasi-linear filtering event has been made. The PSIM module is called with the time interval DELTM and code for the computation of the state transition matrix. Returning to the main program with the state transition matrix, DYNØ is called to compute the dynamic noise. The next sequence of calls in the main program are to the TRAKM, MENØ, and NAVM modules. These three modules return the observation matrix, measurement noise, and covariance matrix as well as the gain matrix.

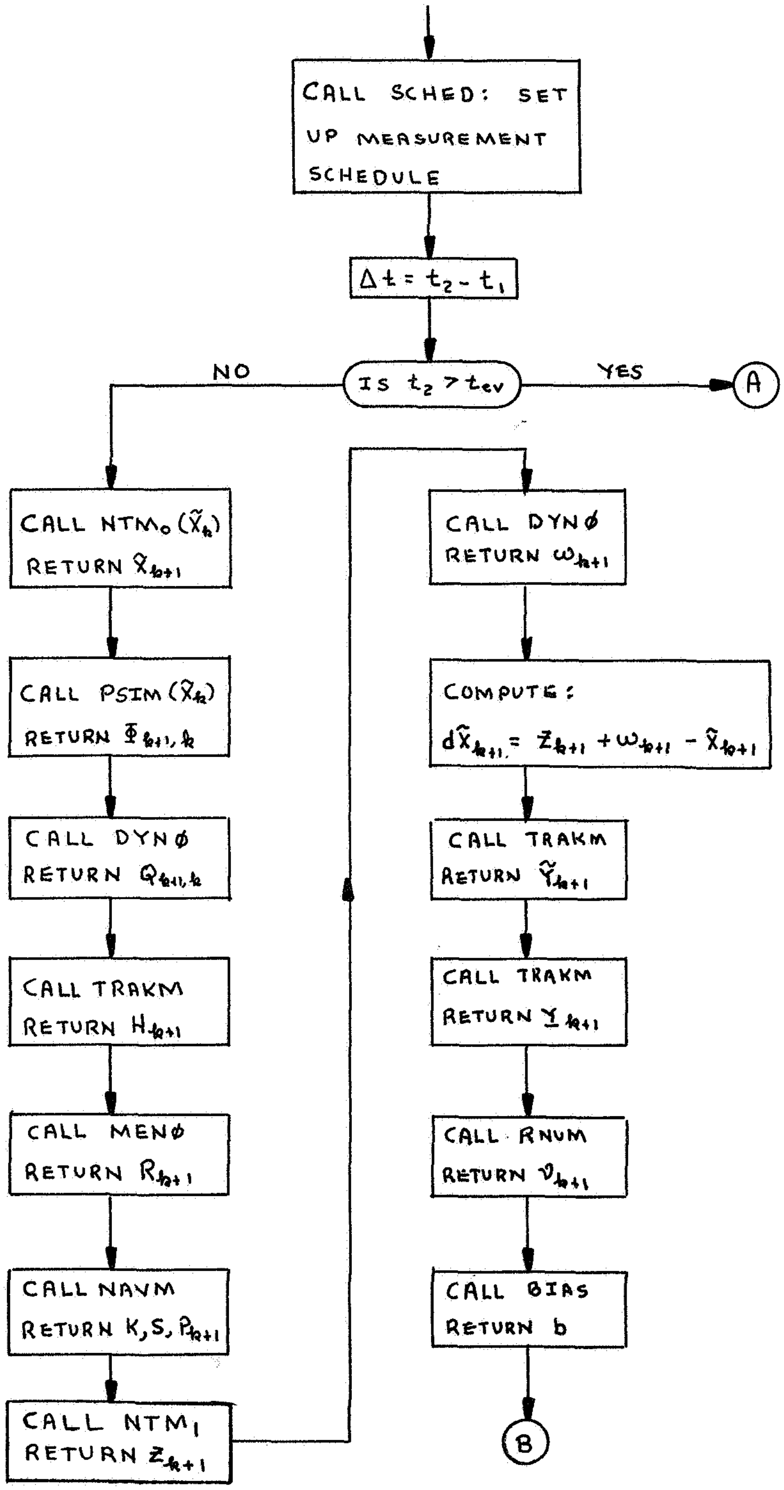


Figure 4.- Simulation Mode Logic

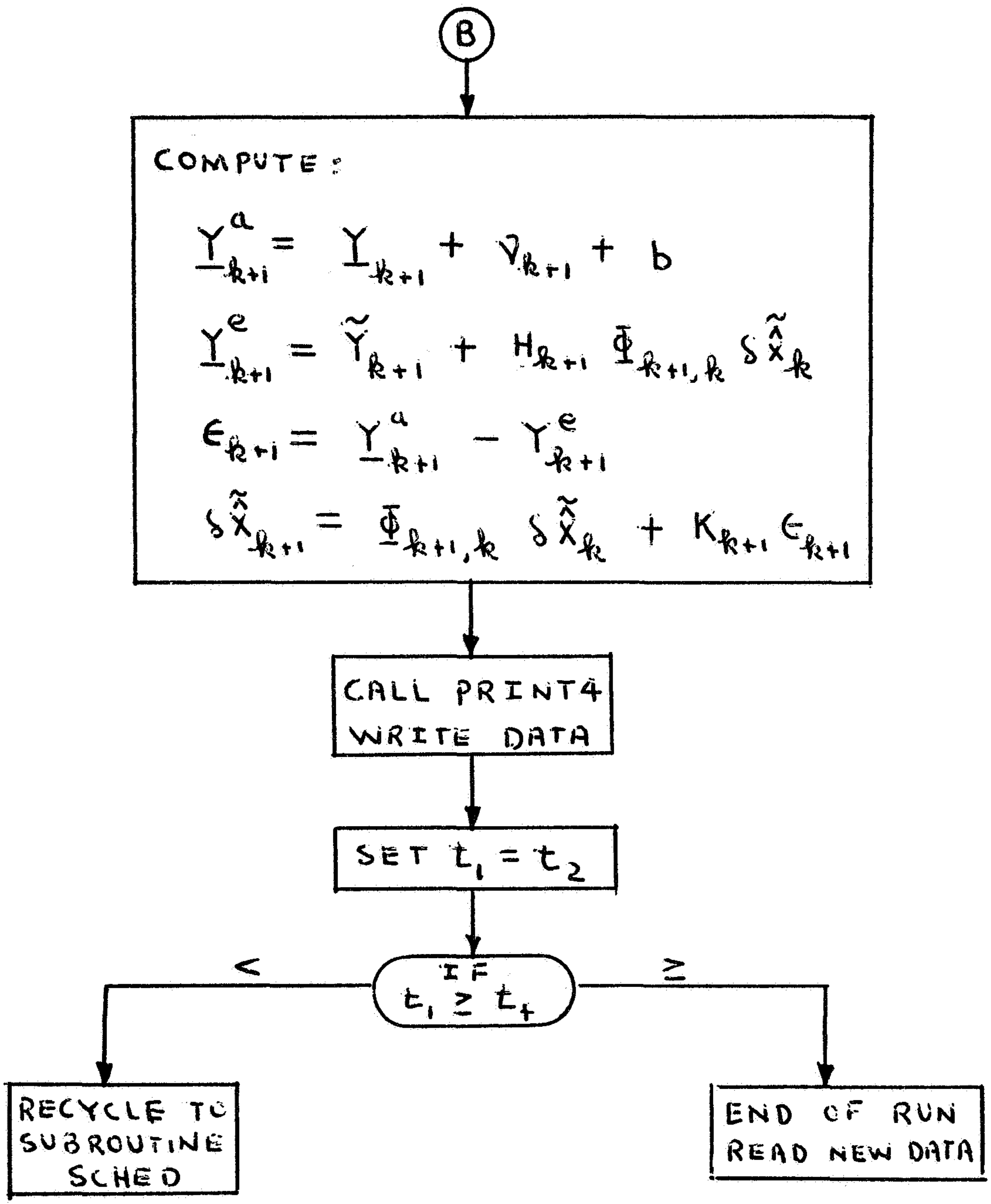


Figure 4.- Continued

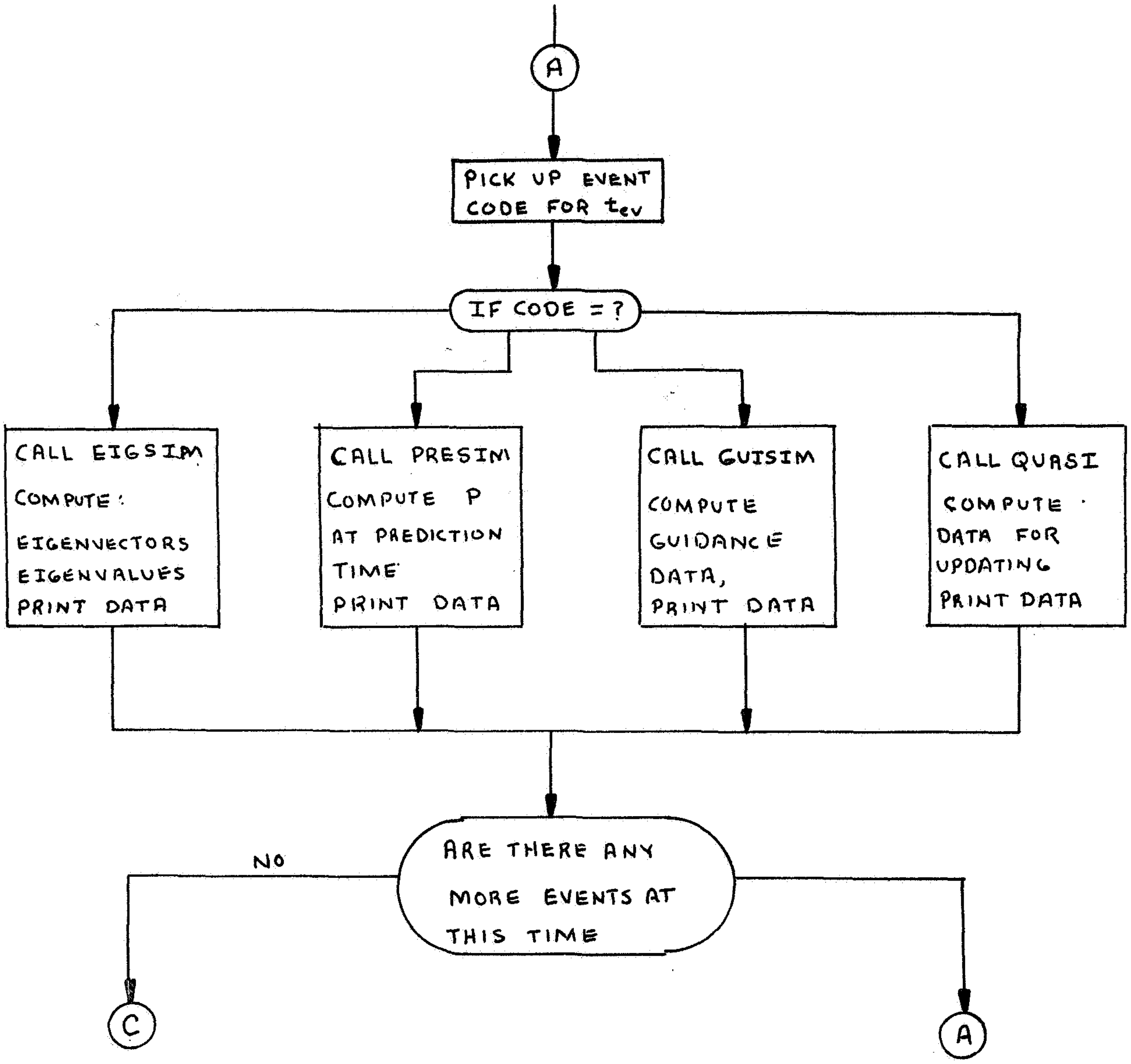


Figure 4.- Concluded

The next module called by the main program is NTM. When NTM is used at this computational time the actual dynamics are used to define the "actual" trajectory. The NTM module computes and returns the "actual" state vector. The DYNØ module is called next, which returns the actual unmodeled accelerations. The actual deviation from the most recent nominal trajectory is then computed in the main program. The TRAKM module is called to calculate and return the most recent nominal observation. This same module is called again, with the "actual" state vector, to compute and return what would have been actually measured if there were no measurement errors. The main program calls the measurement noise subroutine MENØ and computes the white noise matrix corrupting the actual measurement. After returning with the white noise matrix, the random number generator RNUM is called to compute the white noise components. The next computation in the main program calls the subroutine BIAS, which computes the measurement biases. On returning to the main program, the actual measurement, expected measurement, and the actual measurement residuals are computed. Other calculations are performed to obtain the actual orbit determination inaccuracy, actual deviation from the nominal trajectory, and the estimated deviation from the original nominal trajectory. Upon completion of the foregoing computations, the PRINT4 subroutine is called to write out all the desired data. After a subsequent return to the main program the process is repeated by updating DELTM. Recycling is continued until the final trajectory time FNTM is reached, at which time the run is terminated.

If the latest trajectory time TRTM2 has gone past a scheduled event time, the normal mode of operation is interrupted. The scheduled event time and type of event is determined at this point. If a quasi-linear filtering event is to take place, the subroutine QUASI is called. The other possible events, eigenvector, prediction, or guidance would call EIGSIM, PRESIM, and GUISIM, respectively. The computations in QUASI update the nominal trajectory by taking into account the estimated deviations from the most recent nominal trajectory. The other events are identical with those in the error analysis mode. After the necessary computations are made in the event subroutines, control is returned to the main program and the normal processing of information is continued until the final trajectory time is reached. The run is terminated at this point unless additional runs are to be made.

To complete the discussion of the simulation mode logic, comments concerning a quasi-linear filtering event are necessary. This event is determined at the logical IF statement at the

beginning of the run. Recall that in the simulation mode four trajectories are carried along from measurement to measurement: $\bar{X}_k$ the original nominal, $\tilde{X}_k$ the most recent nominal, $d\tilde{X}_k$ the actual deviation from the most recent nominal, and $\delta\hat{X}$ the estimated deviation from the most recent nominal. If a quasi-linear filtering event is to take place, then the original nominal trajectory is updated by using the most recent estimate of the nominal trajectory. Hence the new values of the four trajectories after a quasi-linear filtering event are given by,

$$\bar{X}^{+}_{tev} = \bar{X}^{-}_{tev}$$

$$\tilde{X}^{+}_{tev} = \tilde{X}^{-}_{tev} + \delta\tilde{\hat{X}}^{-}_{tev}$$

$$d\tilde{X}^{+}_{tev} = d\tilde{X}_{tev} - \delta\tilde{\hat{X}}^{-}_{tev}$$

$$\delta\tilde{\hat{X}}^{+}_{tev} = 0$$

It should be noted that if a quasi-linear filtering event has already taken place in the basic cycle, the most recent nominal trajectory has to be computed in the nominal trajectory module. Upon completion of the above computations in subroutine QUASI, control is returned to the basic cycle.

V. MAIN PROGRAM AND SUBROUTINE DESCRIPTIONS

This chapter describes in all necessary detail the formulation, rationale, and computational logic for the routines that make up the entire STEAP. The main program and subroutines are documented in a complete and concise manner so that modifications to the existing routines can be made without much difficulty.

An appreciation for the complete program logic may be gained by a careful review of each subroutine and its corresponding flow chart. If only program utilization is of interest, the user is referred to the example runs in chapter VIII of this volume. In addition, Volume II of this document discusses several numerical examples.

A. MAIN Program (STEAP)

MAIN (Trajectory, Error Analysis, and Simulation Mode)

Purpose: The MAIN routine is the master driver for the entire Simulated Trajectories Error Analysis Program (STEAP). MAIN sets up the necessary linkage to run the three operational modes of the program -- trajectory, error analysis, and simulation. The targeting mode has a separate MAIN program detailed in Section B following.

Calling sequence: None.

Input/output: None.

Subprograms required:

BIAS	MENO	PRINT3	SCHED
DATA	NAVM	PRINT4	TRAKM
DYNO	NTM	PRNTS3	
EIGEN	PRED	PRNTS4	
EIGSIM	PRESIM	PSIM	
GUIDM	PRINT1	QUASI	
GUISIM			

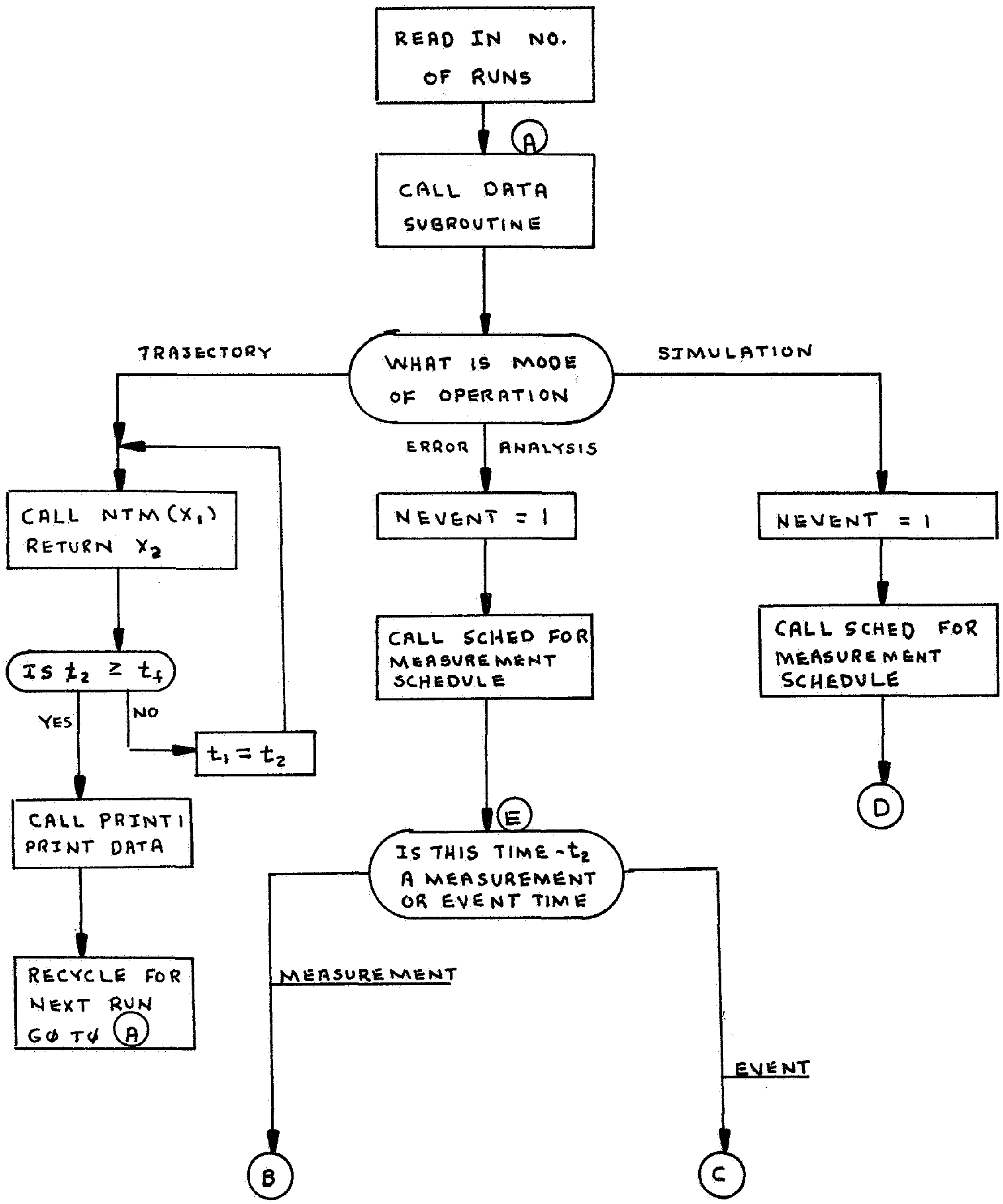
Approximate storage required (OCTAL): 3204.

Discussion: In designing STEAP, flexibility and computational speed were the prime factors. The program was designed in block modules that have access to all the subroutines within STEAP. In this manner any subroutine or module can be inserted or replaced by the user. Hence, the MAIN routine is relatively simple in that each operational mode (three available) is a designated block within MAIN. The three blocks are targeting, error analysis, and simulation.

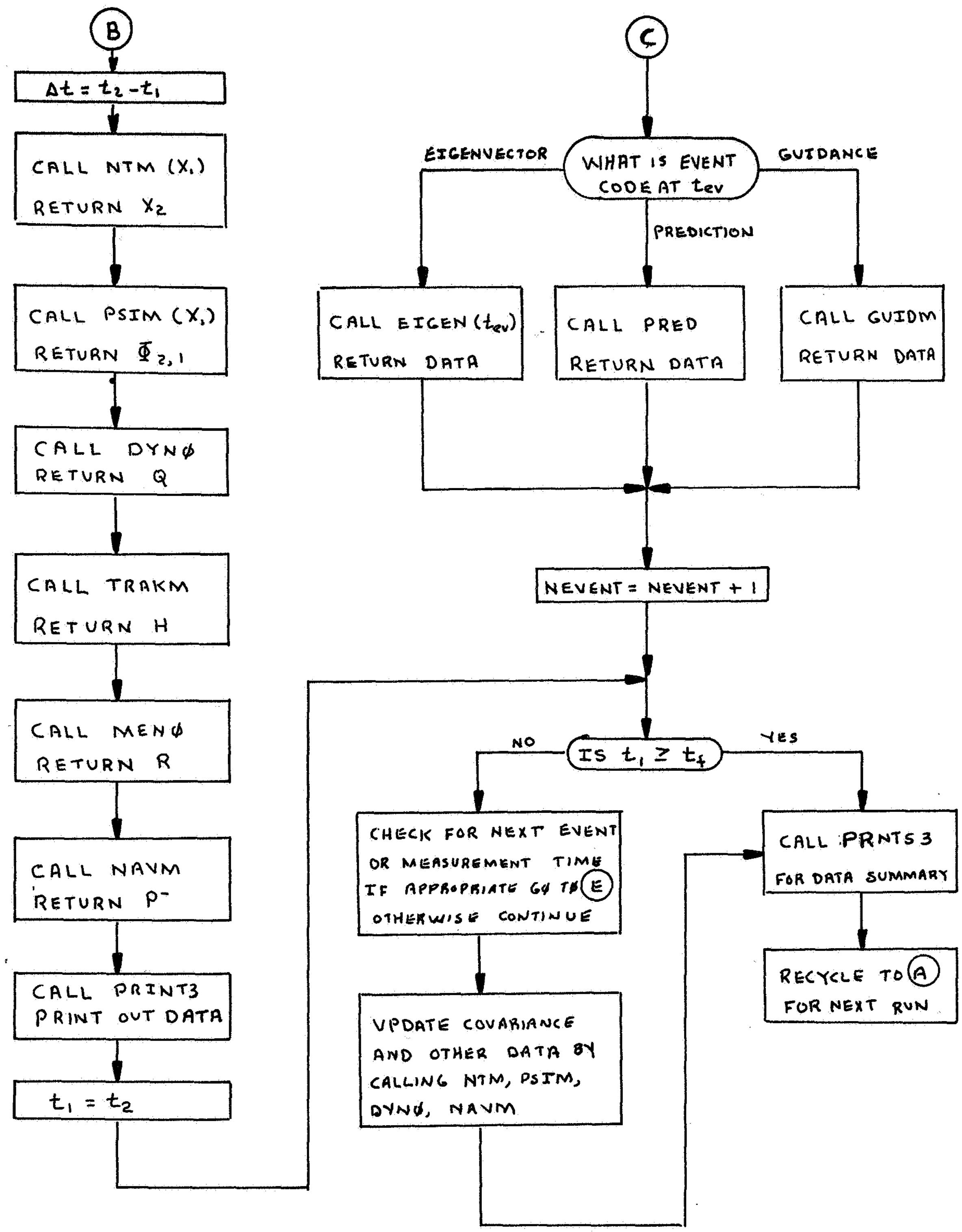
The first operation performed by MAIN reads in a data card that signifies the number of runs to be made. With each succeeding run the MAIN routine calls the DATA subroutine, which provides all the necessary data for each mode of operation. MAIN then proceeds to the module for which the program is to be exercised and performs the necessary computational logic. Control is always returned to MAIN after each run.

The MAIN routine computational logic follows this discussion.

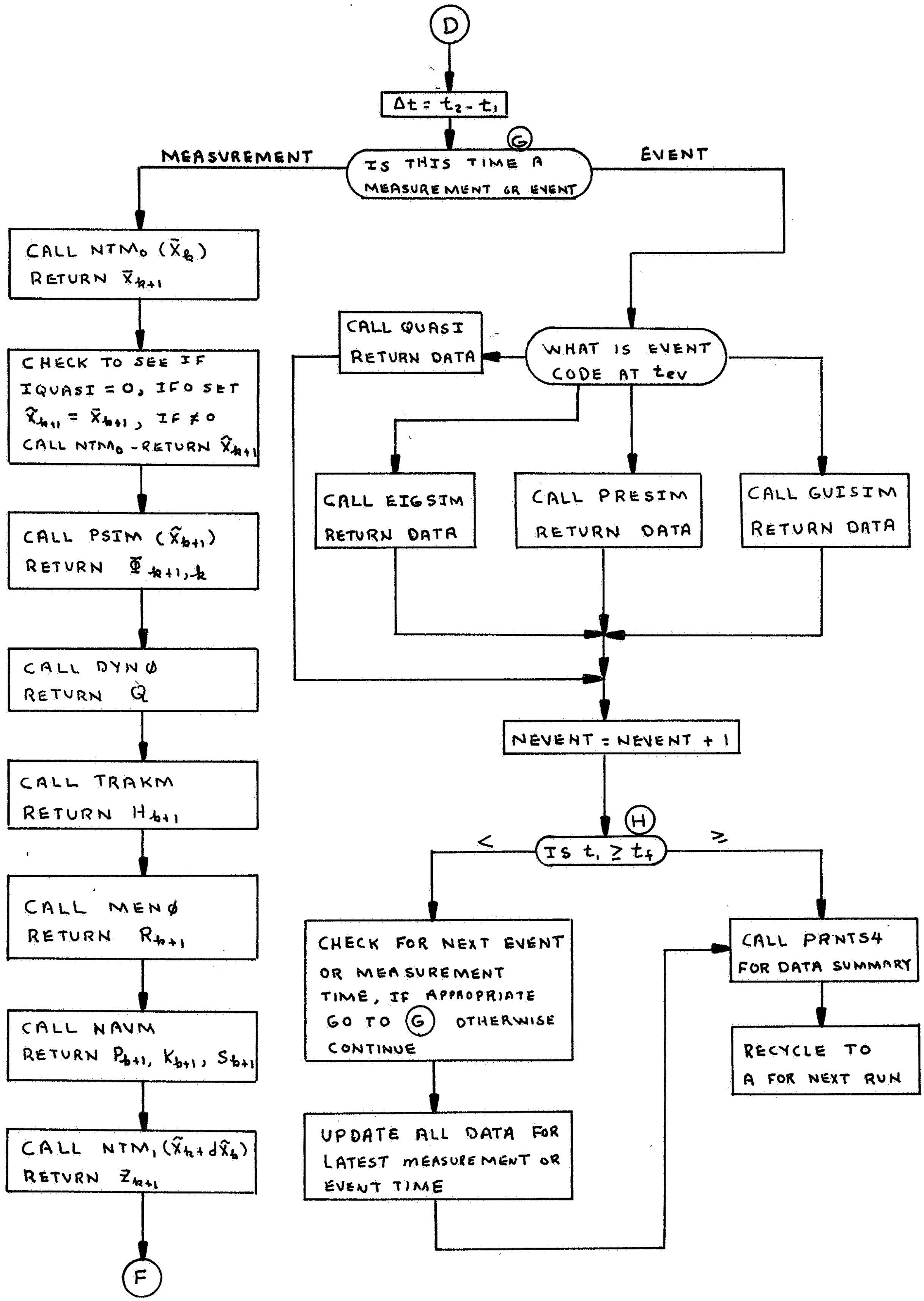
Computational logic:



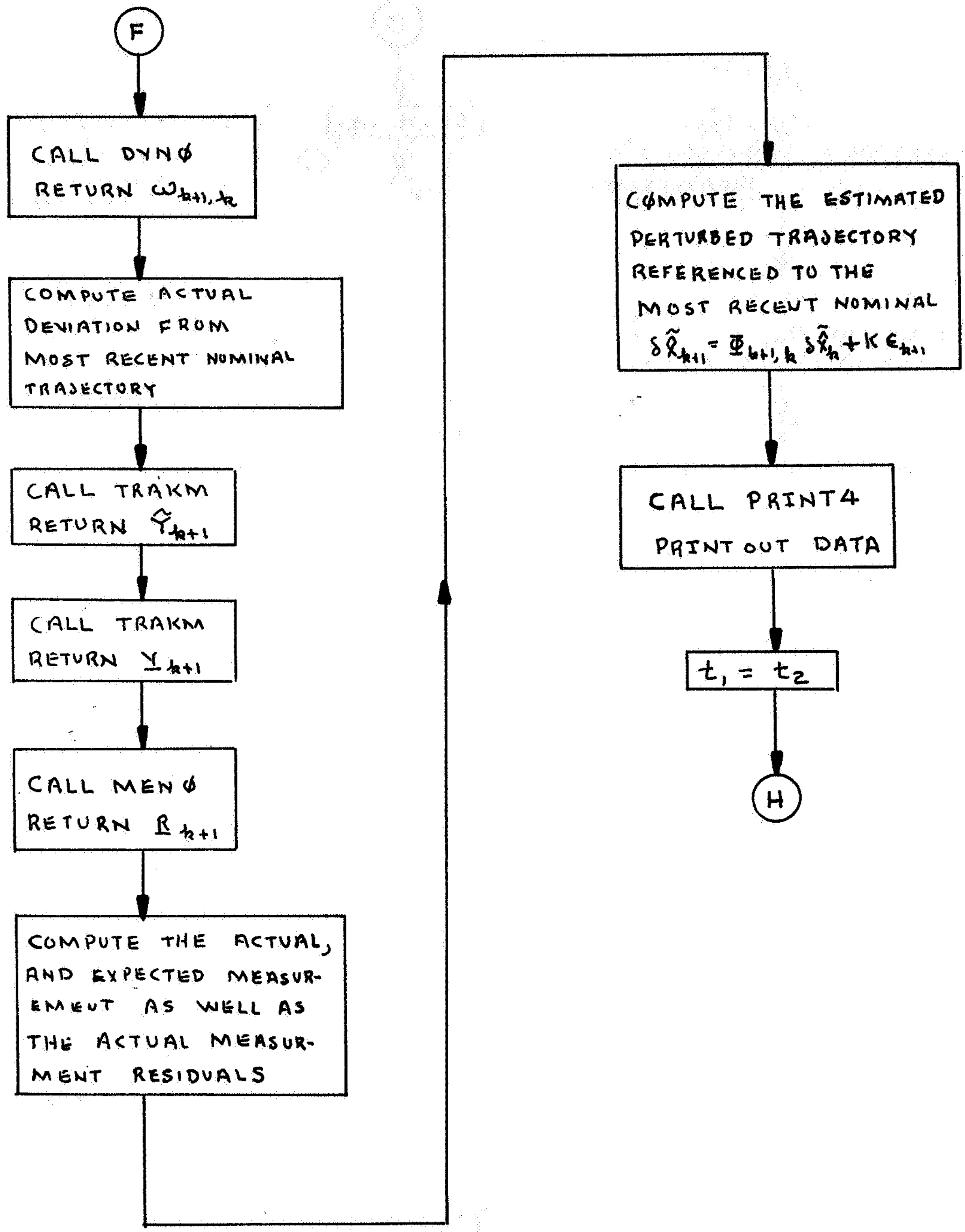
Computational logic (continued):



Computational logic (continued):



Computational logic (concluded):



B. MAIN Program (Targeting)

MAIN (Targeting Mode)

Purpose: This program controls the operation of the entire targeting mode.

Calling sequence: None.

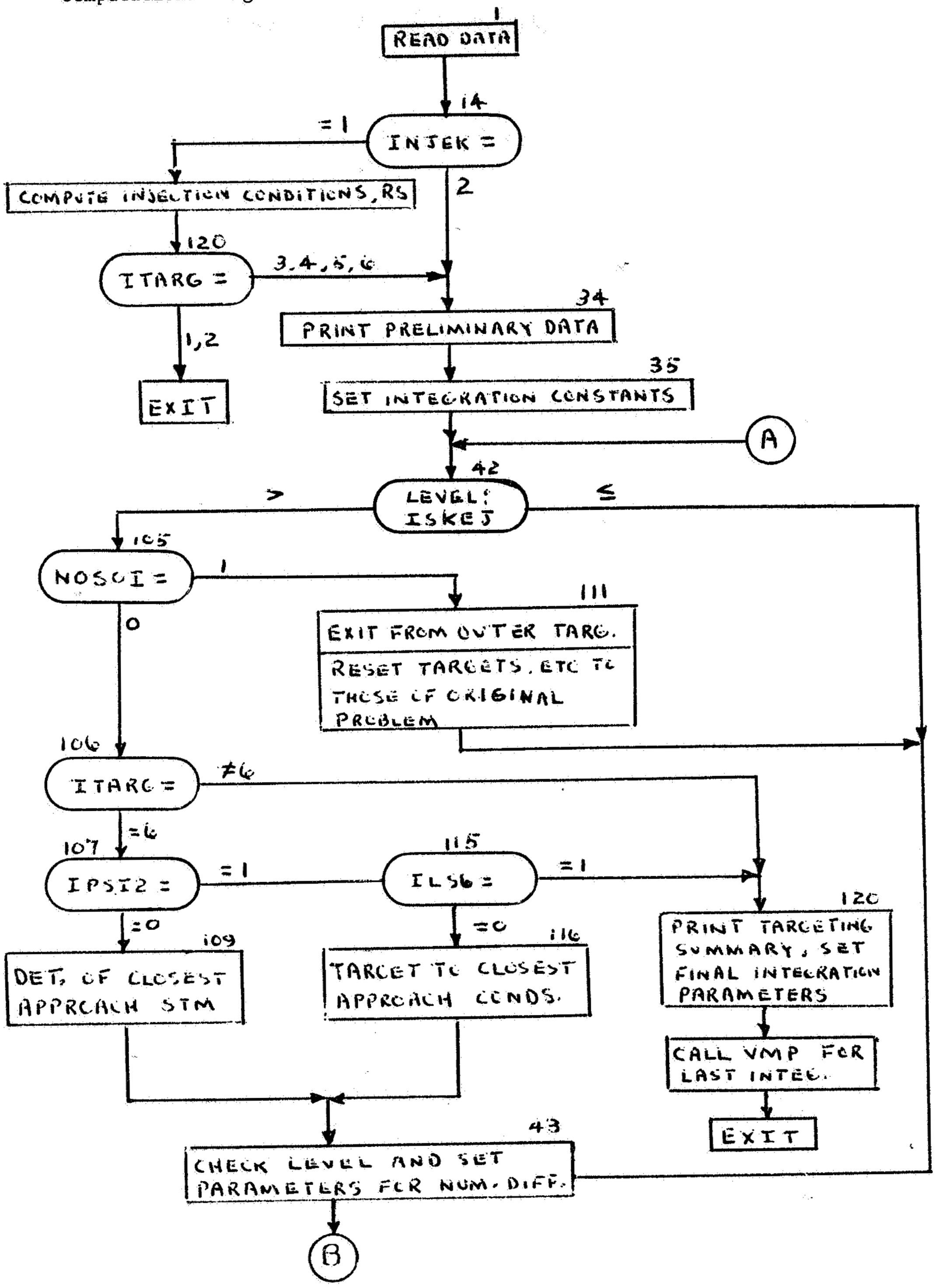
Input/output: Input/output options are discussed in detail in Chapter II.B and III.B.

Subprograms required:

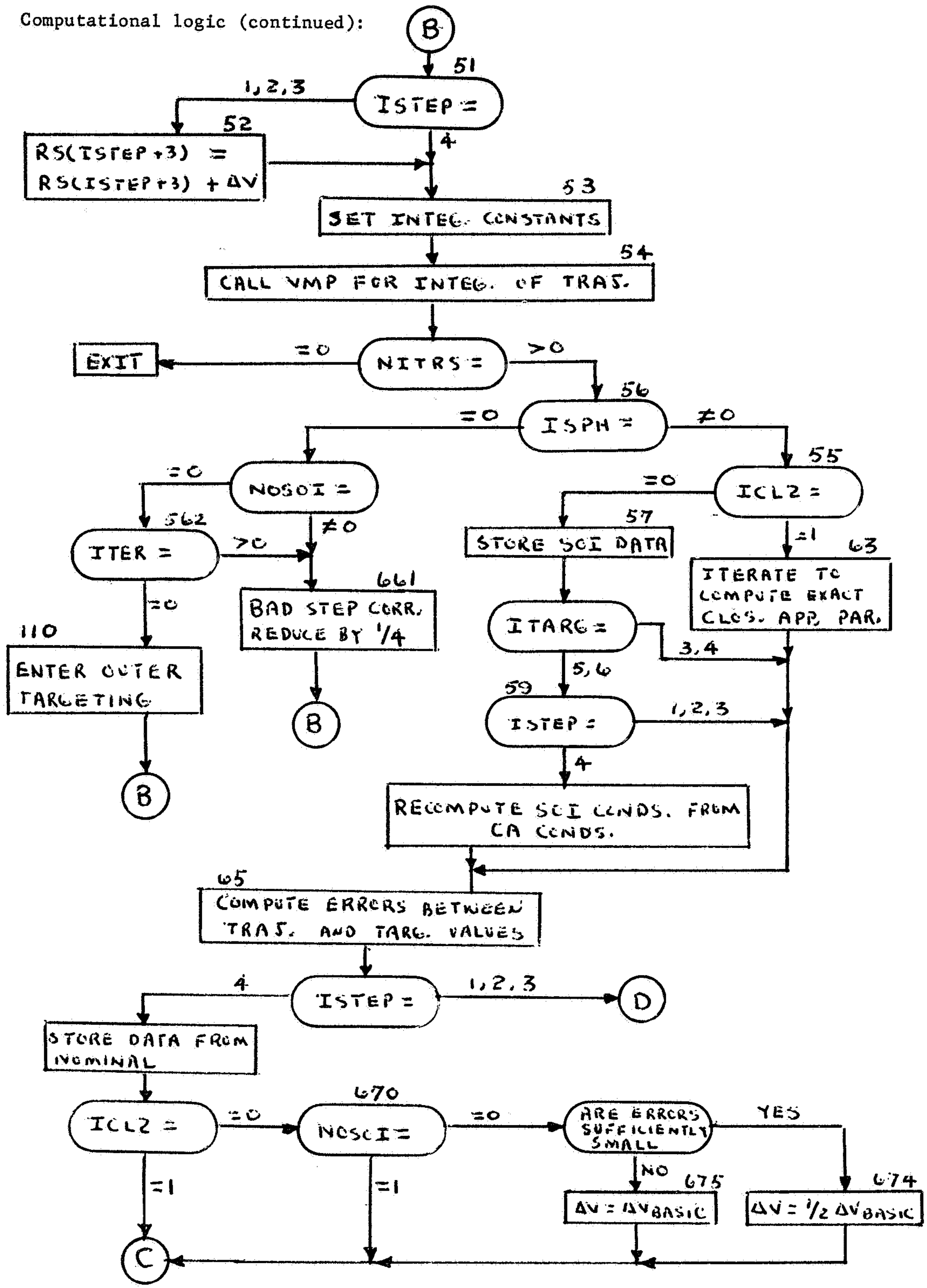
ACTB	ESTMT	NEWPAGE	POSVL
AUX	EULMX	NJEXN	PRINT
BLKDAT	HYPER	ORB	SPACE
CASOI	HYPSV	OT2	TIME
CONIC	INPUTZ	OUT1	VECTOR
CONST	LAMB	PECEQ	VMASS
EPHEM	MATIN	PLANE	VMP

Discussion: The program is described in Chapter IV.B. A detailed flow chart is given on the following pages.

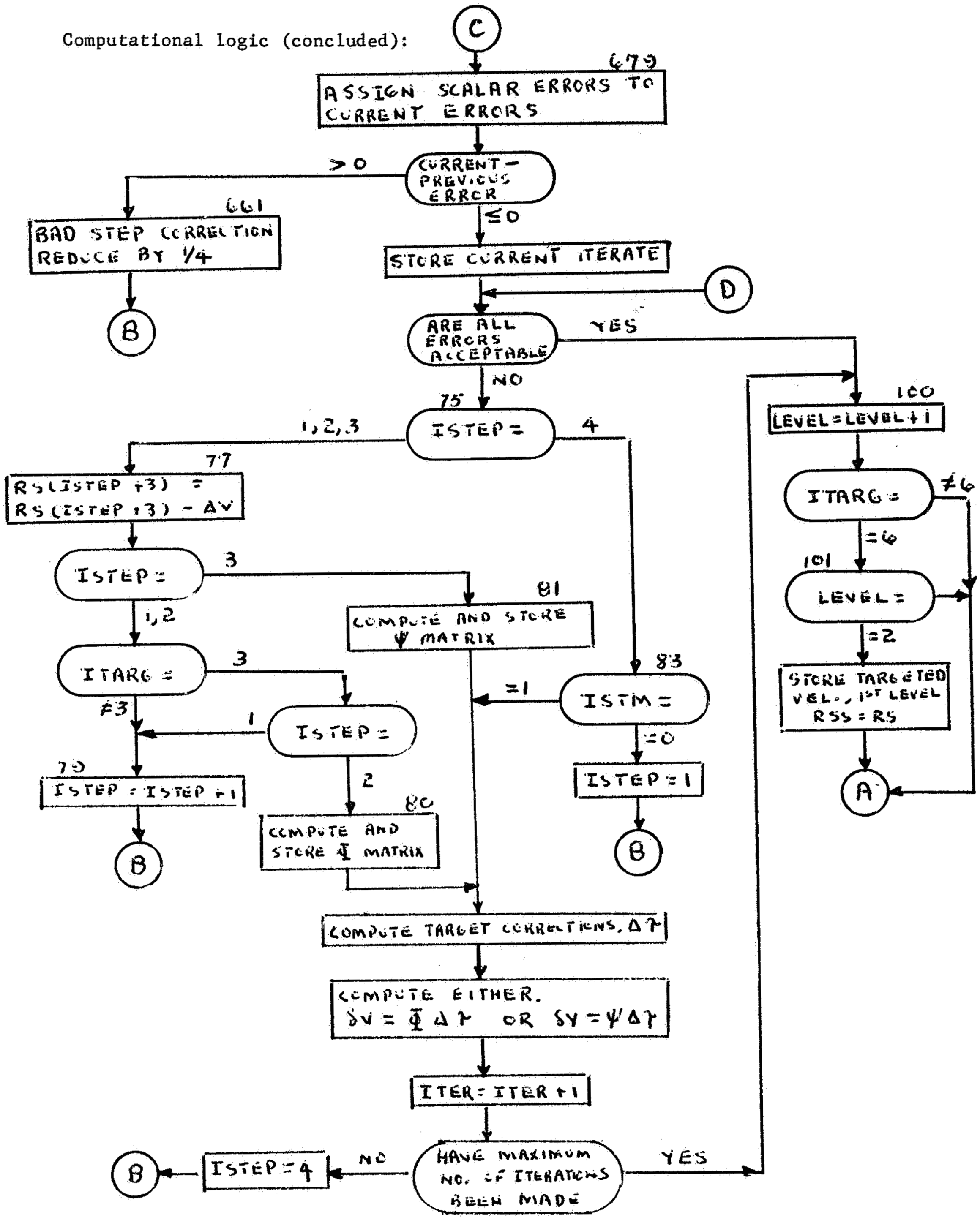
Computational logic:



Computational logic (continued):



Computational logic (concluded):



C. Subroutines

1. Subroutine ACTB

Purpose: Given the gravitational constant of a body and the influence of position and velocity of a vehicle when it enters the sphere of that body, this routine computes the magnitudes of B, B•T, and B•R.

Calling sequence: CALL ACTB(R, V, GMX, B, BDT, BDR)

Input/output:

I/O	Fortran name	Math symbol	Definition
I	R(3)	$\vec{r}$	Position of vehicle relative to the body
I	V(3)	$\dot{\vec{r}}$	Velocity of vehicle relative to the body
I	GMX	μ	Gravitational constant of the body
O	B	B	B-plane coordinates
O	BDT	B•T	
O	BDR	B•R	

Subprograms required: None.

Approximate storage required (OCTAL): 460.

Discussion: The position and velocity of the vehicle relative to the planet are used to compute the elements that define the hyperbola about the planet. The standard coordinate system R, S, T is then constructed. B is defined as that vector lying in the B-plane which extends from the center of the planet to the approach asymptote. Finally the values B•T and B•R are computed.

2. Subroutine AUX

Purpose: This program is responsible for the calculation of the SPARC injection conditions printed out in the point-to-point conditions.

Calling sequence: CALL AUX (W, ELAT, ELON, AZ, PV, Q, TAI, ANG1, ANG2, T1M1, T1M2, S, E, RP, GNE, ROT, DJL, TL, TB, PHI, THI, RAI, AZI, T1NJ, TC).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	W(3)	$\hat{W}$	Normal to launch plane
I	ELAT	φ_L	Latitude of launch site
I	ELON	θ_L	Longitude of launch site
I	AZ	Σ	Desired launch azimuth
I	PV(3)	$\hat{P}$	Unit vector in direction of periapsis of hyperbola
I	Q(3)	$\hat{Q}$	Unit vector in plane of hyperbola perpendicular to PV
I	TAI	ν_I	True anomaly at injection
I	ANG1	ψ_1	Angle of first burn
I	ANG2	ψ_2	Angle of second burn
I	T1M1	t_2	Time of first burn, sec
I	T1M2	t_1	Time of second burn, sec
I	S	$\hat{S}$	Unit vector in direction of departure asymptote
I	E	e	Eccentricity of hyperbola
I	RP	r_p	Periapsis radius of hyperbola
I	GME	μ_p	Gravitational constant of launch planet
I	ROT	----	Rotational rate of launch planet
I	DJL	----	Julian date of launch
O	TL	T_L	Time (hr) of launch after zero hours on date of launch

0	TB	T_B	Time between launch and injection
0	PHI	Φ_I	Injection latitude
0	THI	Θ_I	Injection longitude
0	RAI	Ω_I	Injection right ascension
0	AZI	Σ_I	Injection azimuth
0	T1NJ	----	Time (hr) of injection from zero hours on date of launch
0	TC	T_c	Coast time

Subprograms required: None.

Approximate octal required: 700.

Discussion: The computations in this program are quite elementary. A review of the program listing is sufficient for understanding the program.

3. Subroutine BIAS

Purpose: This subroutine determines which type of measurement is being taken and returns the actual measurement bias to be used in the simulation mode.

Calling sequence: CALL BIAS (MMCODE, BVAL).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	MMCODE	----	A code describing which type of measurement is being taken
O	BVAL(4)	----	The actual bias to be used in the measurement

Subprograms required: None.

Approximate storage required (octal): 50.

Discussion: A vector BIA(12) is input that determines the actual bias to be used in any given measurement. This vector is described in more detail in Chapter II, Input Options. After deciding what parameters are being measured, the appropriate values are placed in the vector BVAL to be returned to the simulation mode. The length of BVAL may vary from one value to three values according to which measurement is being taken.

4. Subroutine BLOCK DATA

Purpose: This subroutine contains the constants used in various other parts of the program.

Calling sequence: None.

Input/output: None.

Subprograms required: None.

Approximate storage required (octal): 10.

Discussion: No computations are accomplished in this routine. The constants mentioned above are loaded with the rest of the program and are ready for use immediately.

5. Subroutine CASOI

Purpose: This program converts closest approach target conditions to sphere of influence conditions.

Calling sequence: CALL CASOI (RS, VHP, TTG, EQEC, DINCL, DRCA, DB, DBDT, DBDR, TSICA)

Input/output:

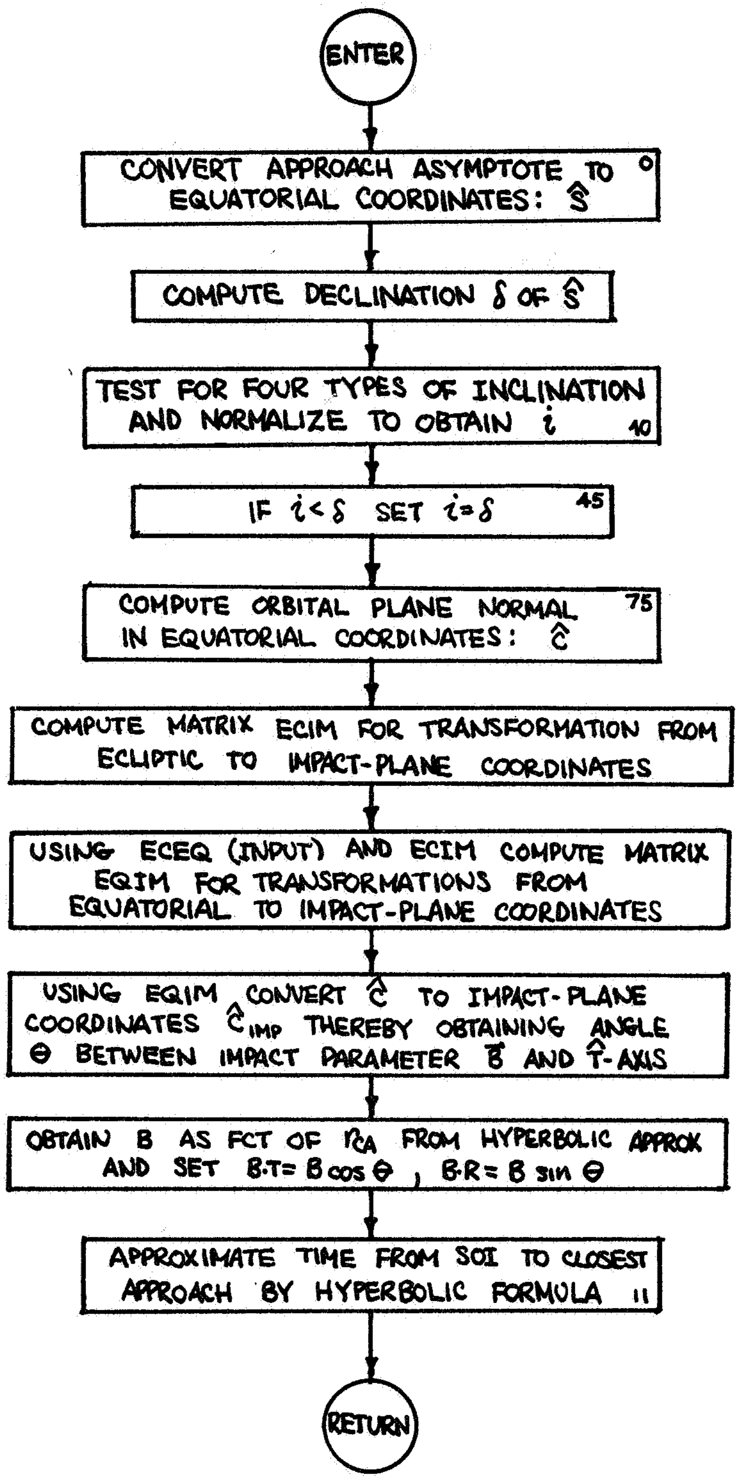
I/O	Fortran name	Math symbol	Definition
I	RS(3)	r_{SOI}	Position vector at sphere of influence (planet-centered ecliptic)
I	VHP(3)	V_{SOI}	Velocity vector at sphere of influence (planet-centered ecliptic)
I	TTG	μ_P	Gravitational constant of target planet
I	EQEC(3, 3)	M_{EQEC}	Transformation matrix from equatorial to ecliptic coordinates
I	DINCL	i_{CA}	Inclination at closest approach, rad
I	DRCA	r_{CA}	Radius at closest approach
O	DB	B	Impact parameter
O	DBDT	B·T	Impact parameter variable
O	DBDR	B·R	Impact parameter variable
O	TSICA	Δt	Time from sphere of influence to closest approach, days

Subprogram required: None.

Approximate storage required (octal): 1500.

Discussion: The program determines the plane of motion about the target planet from the target inclination and approach asymptote. The normal to the plane of motion projected on the impact plane determines the angle between the T-axis and the miss vector B. B·T and B·R are calculated from this angle. The time from the sphere of influence to closest approach is based on a patched conic approximation.

Computational logic:



6. Subroutine CONC2

Purpose: This subroutine computes the state transition matrix of dimension 6 x 6 that relates perturbations about a nominal trajectory at the times t_{k+1} and t_k.

Calling sequence: CALL CONC2 (R, V, DELT, GMX, PSIEC).

Input/output

I/O	Fortran name	Math symbol	Definition
I	R(3)	$\vec{r}$	Position of the vehicle relative to the governing body
I	V(3)	$\dot{\vec{r}}$	Velocity of the vehicle relative to the governing body
I	DELT	Δt	Time increment over which the state transition matrix is being computed
I	GMX	μ	Gravitational constant of the governing body
O	PSIEC(6,6)	Φ	State transition matrix

Subprograms required: None.

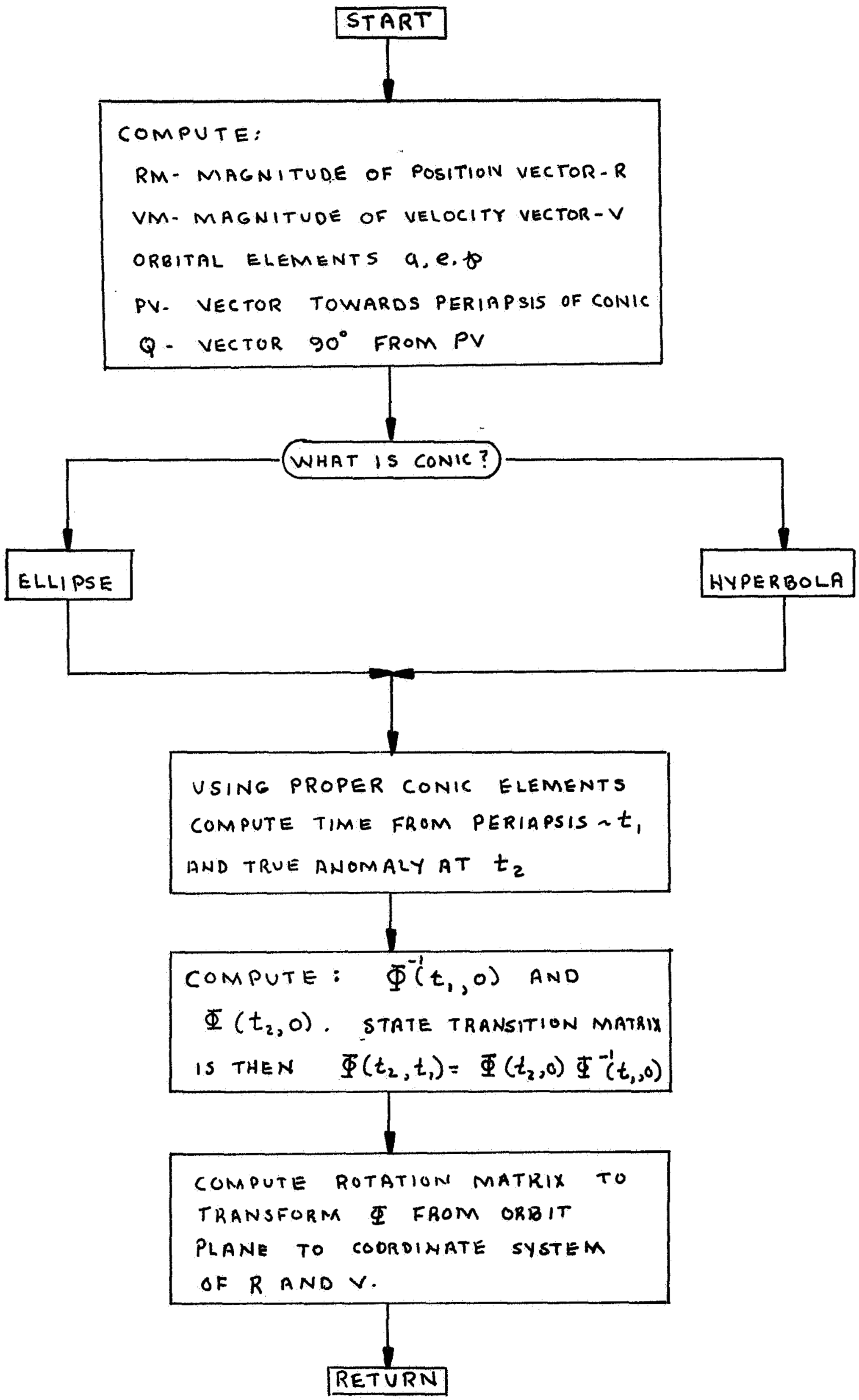
Approximate storage required (octal): 1550.

Discussion: State transition matrices are used to find small departures in position and velocity from Keplerian motion at two different times, t_{k+1} and t_k. In using analytical methods to determine Φ, the assumption is made that over a small time interval of an interplanetary trajectory, the motion of the vehicle is essentially a two-body conic section.

Computation of state transition matrices in this subroutine is based on the method as given by Danby (ref. 2). A discussion of this technique is presented in the analytical manual, Volume II of this report.

CONC2 is used for both virtual mass and patched conic state transition matrix computations. The difference lies in the use of the gravitational parameter μ. When computing Φ from virtual mass concepts, the μ of the effective force center is used. When patched conic methods are used, the μ of the dominant body is considered in the analysis. The method of how Φ is computed is determined by input.

Computational logic:



7. Subroutine CONIC

Purpose: This program determines the elements of the ellipse specified by given position and velocity vectors.

Calling sequence: CALL CONIC (R, RM, V, VM, A, E, XI, XL, W, TA, PV, Q, GMX, RP, P, WV).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	R(3)	$\vec{r}$	Position vector
I	RM	r	Position vector magnitude
I	V(3)	$\vec{v}$	Velocity vector
I	VM	v	Velocity vector magnitude
0	A	a	Semi-major axis
0	E	e	Eccentricity
0	XI	i	Inclination
0	XL	Ω	Longitude of ascending node
0	W	ω	Argument of periapsis
0	TA	ν	True anomaly of specified position
0	PV(3)	$\hat{P}$	Standard unit vecotor in direction of periapsis
0	Q(3)	$\hat{Q}$	Standard unit vector in orbital plane normal to PV
I	GMX	μ_P	Gravitational constant of primary body
0	RP	Ω_P	Periapsis radius
0	P	P	Semi-latus rectum
0	WV	W	Normal to orbital plane

Subprograms required: None.

Approximate storate required (octal): 400.

Discussion: This is a standard conic section program.

8. Subroutine CONST

Purpose: This program sets the launch profile constants used in the NJEXN subroutine.

Calling Sequence: CALL CONST (NDD, NTT, RP, HHTA, ANG1, ANG2, TIM1, TIM2, DDLAT, DDLON, DDIQ, DDLQ, ROT).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	NDD		Index of launch planet
I	NTT		Index of target planet
0	RP	r_P	Parking orbit radius
0	HHTA	ν_I	True anomaly at injection
0	ANG1	Ψ_1	Angle of first burn
0	ANG2	Ψ_1	Angle of second burn
0	TIM1	t_1	Time of first burn
0	TIM2	t_2	Time of second burn
0	DDLAT	ϕ_L	Latitude of launch site
0	DDLON	θ_L	Longitude of launch site
0	DDIQ		Obliquity of launch planet orbit
0	DDLQ		Ascending node of launch planet orbit
0	ROT		Rotational rate of launch planet

Subprograms required: None.

Approximate storate required (octal): 300.

Discussion: The output parameters are simply set equal to desired values.

9. Subroutine CONVERT

Purpose: If at a given time the geocentric radius, declination, right ascension, velocity, path angle, and azimuth of a vehicle are known, this subroutine will calculate the geocentric equatorial coordinates of the vehicle.

Calling sequence: CALL CONVERT (R, PHI, THETA, VEL, GAMMA, SIGMA, X, Y, Z, VX, VY, VZ).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	R	r	Geocentric radius
I	PHI	ϕ	Declination
I	THETA	θ	Right ascension
I	VEL	v	Velocity
I	GAMMA	γ	Path angle
I	SIGMA	σ	Azimuth
The following coordinates are geocentric equational			
0	X	x	Position coordinates
0	Y	y	
0	Z	z	
0	VX	$\dot{x}$	Velocity coordinates
0	VY	$\dot{y}$	
0	VZ	$\dot{z}$	

Subprograms required: None.

Approximate storage required (octal): 170.

Discussion: The following formulas are used to find the geocentric equatorial coordinates from the given data in this routine:

$$x = r \cos \phi \cos \theta$$

$$y = r \cos \phi \sin \theta$$

$$z = r \sin \phi$$

$$\dot{x} = v (\sin \gamma \cos \phi \cos \theta - \cos \gamma \sin \sigma \sin \theta - \cos \gamma \cos \sigma \sin \phi \cos \theta)$$

$$\dot{y} = v (\sin \gamma \cos \phi \sin \theta + \cos \gamma \sin \sigma \cos \theta - \cos \gamma \cos \sigma \sin \phi \sin \theta)$$

$$\dot{z} = v (\sin \gamma \sin \phi + \cos \gamma \cos \sigma \cos \phi)$$

10. Subroutine DATA

Purpose: This subroutine is responsible for reading the data and translating that data into symbols compatible with the rest of the program.

Calling sequence: CALL DATA.

Input/output: All communication with the subroutine DATA is accomplished through the use of COMMON blocks. Thus, no "dummy arguments" appear in the calling sequence.

Subprograms required: CONVERT, EPHEM, GHA, ORB, TIME, TRANS.

Approximate storage required (octal): 6530.

Discussion: To determine the exact means of reading data refer to Chapter II, Input Options. For those variables for which it is necessary, this subroutine sets initial values before transferring control to the main program. In addition, DATA prints the initial conditions of most variables included in the namelists.

11. Subroutine DYNO

Purpose: The dynamic noise matrix for the error analysis or simulation mode is computed. In addition, DYNO computes the actual dynamic noise used in the simulation mode.

Calling sequence: CALL DYNO (ICODE).

Input/output:

I/O	Fortran name	Definition
I	ICODE	An internal code that determines if the dynamic noise matrix is computed or the actual dynamic noise is calculated.

Subprograms required: None.

Approximate storage required (octal): 260.

Discussion: If ICODE = 0, the dynamic noise matrix is computed as a function of the input vector DNCN in the following manner:

$$Q(1,1) = 0.25(\Delta t)^4 \ \mathrm{DNCN}(1)$$

$$Q(2,2) = 0.25(\Delta t)^4 \ \mathrm{DNCN}(2)$$

$$Q(3,3) = 0.25(\Delta t)^4 \ \mathrm{DNCN}(3)$$

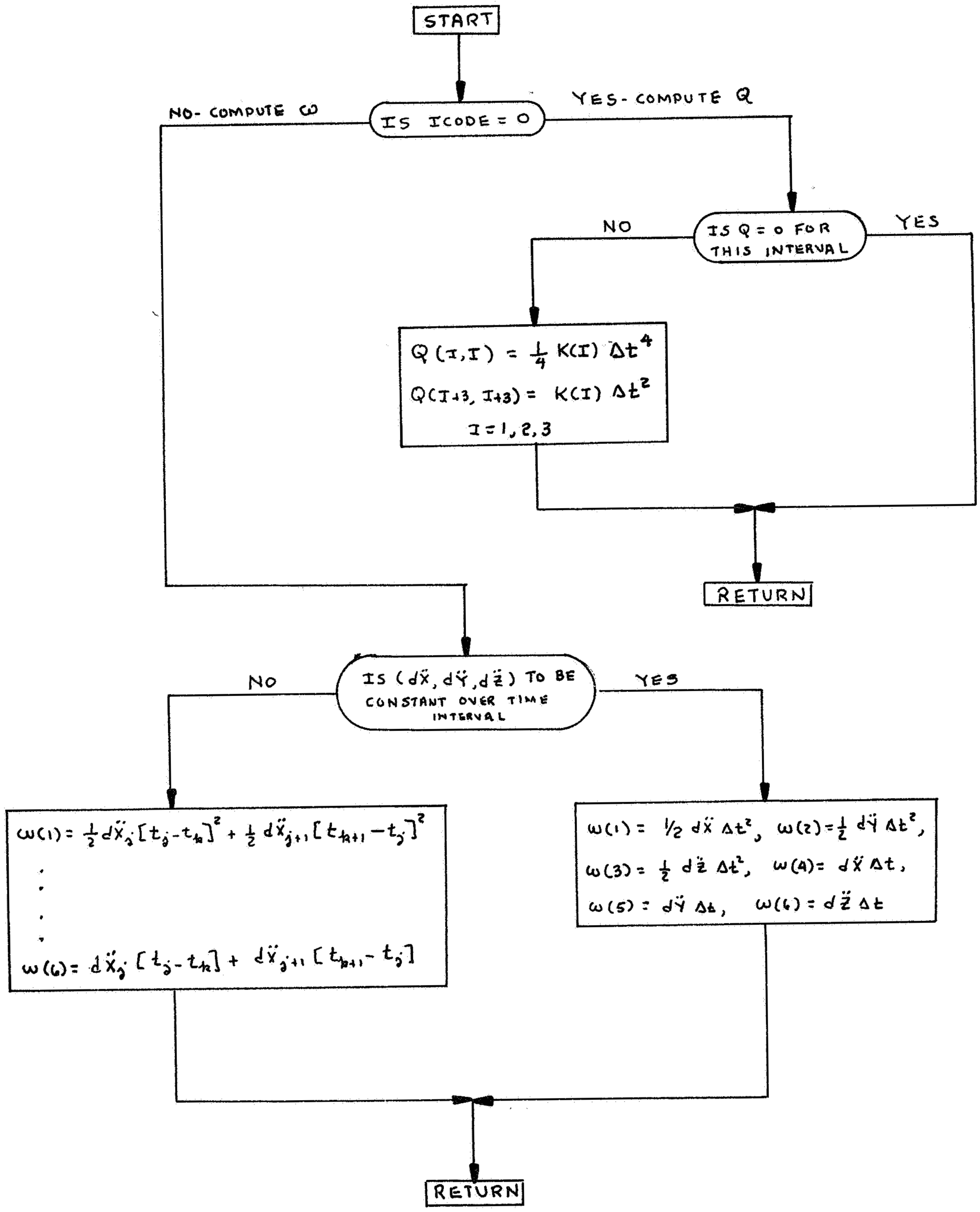
$$Q(4,4) = (\Delta t)^2 \ \mathrm{DNCN}(1)$$

$$Q(5,5) = (\Delta t)^2 \ \mathrm{DNCN}(2)$$

$$Q(6,6) = (\Delta t)^2 \ \mathrm{DNCN}(3)$$

The actual dynamic noise is computed if ICODE = 1. The actual unmodeled acceleration may be input. It is possible to allow a different value of the unmodeled acceleration for each of three different time intervals along the trajectory. The result is stored in the vector W.

Computational logic:



12. Subroutine EIGEN

Purpose: This routine is called on to calculate the eigenvalues, eigenvectors, and hyperellipsoids of the covariance matrix at a previously specified time which is known as the time of an eigenvector event in the error analysis mode.

Calling sequence: CALL EIGEN (RI, TEVN).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	The state vector describing the position and velocity of the vehicle at the time of the last measurement or event
I	TEVN	t_{ev}	The time at which the eigenvector event is to take place

Subprograms required: DYNO, HYELS, JACOBI, NAVM, NTM, PSIM.

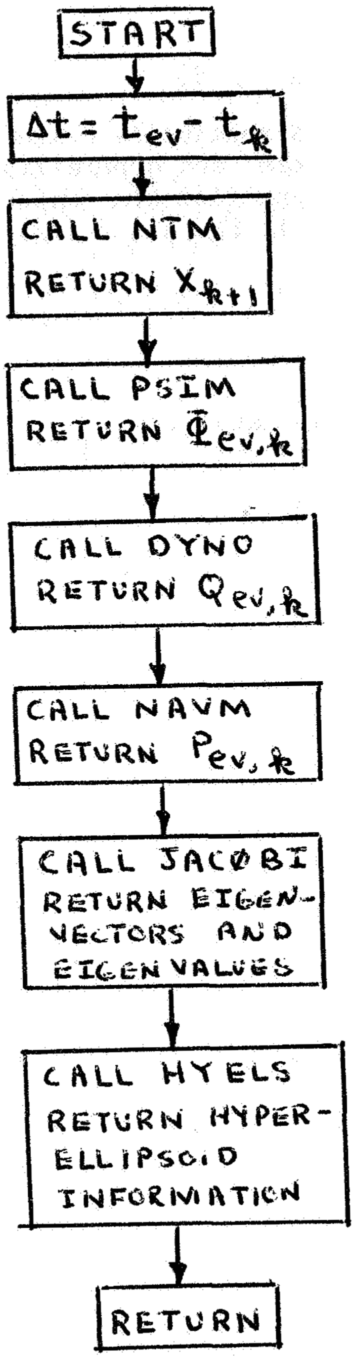
Approximate storage required (octal): 2530.

Discussion: The covariance matrix P is propagated forward from the time of the last measurement or event, t_{k-1}, through the formula

$$P_{t_{ev}, t_{k-1}} = P_{ev, k-1} = \Phi_{ev, k-1} P_{k-1, k-1}, \Phi^T_{ev, k-1} + Q_{ev, k-1}$$

where $\Phi_{ev, k-1}$ is the state transition matrix relating deviations in the state vector at t_{ev} to deviations at t_{k-1}, $Q_{ev, k-1}$ is the dynamic noise matrix at time t_{ev}, and $P_{k-1,k}$ is the covariance matrix at the time of the last measurement or event. The position and/or velocity eigenvalues and eigenvectors and related hyperellipsoids are then computed and printed. The subroutine then returns $P_{ev, k-1} = P_{ev, ev}$ and the state vec-toc at time t_{ev} to the basic cycle in order to process the next measurement or event.

Computational logic:



13. Subroutine EIGSIM

Purpose: The purpose of this subroutine is to obtain the information necessary for an eigenvector event in the simulation mode of STEAP; that is, to compute the eigenvalues and eigenvectors of the covariance matrix, P, at a given time.

Calling sequence: CALL EIGSIM (RI, TEVN, RI1).

Input/output:

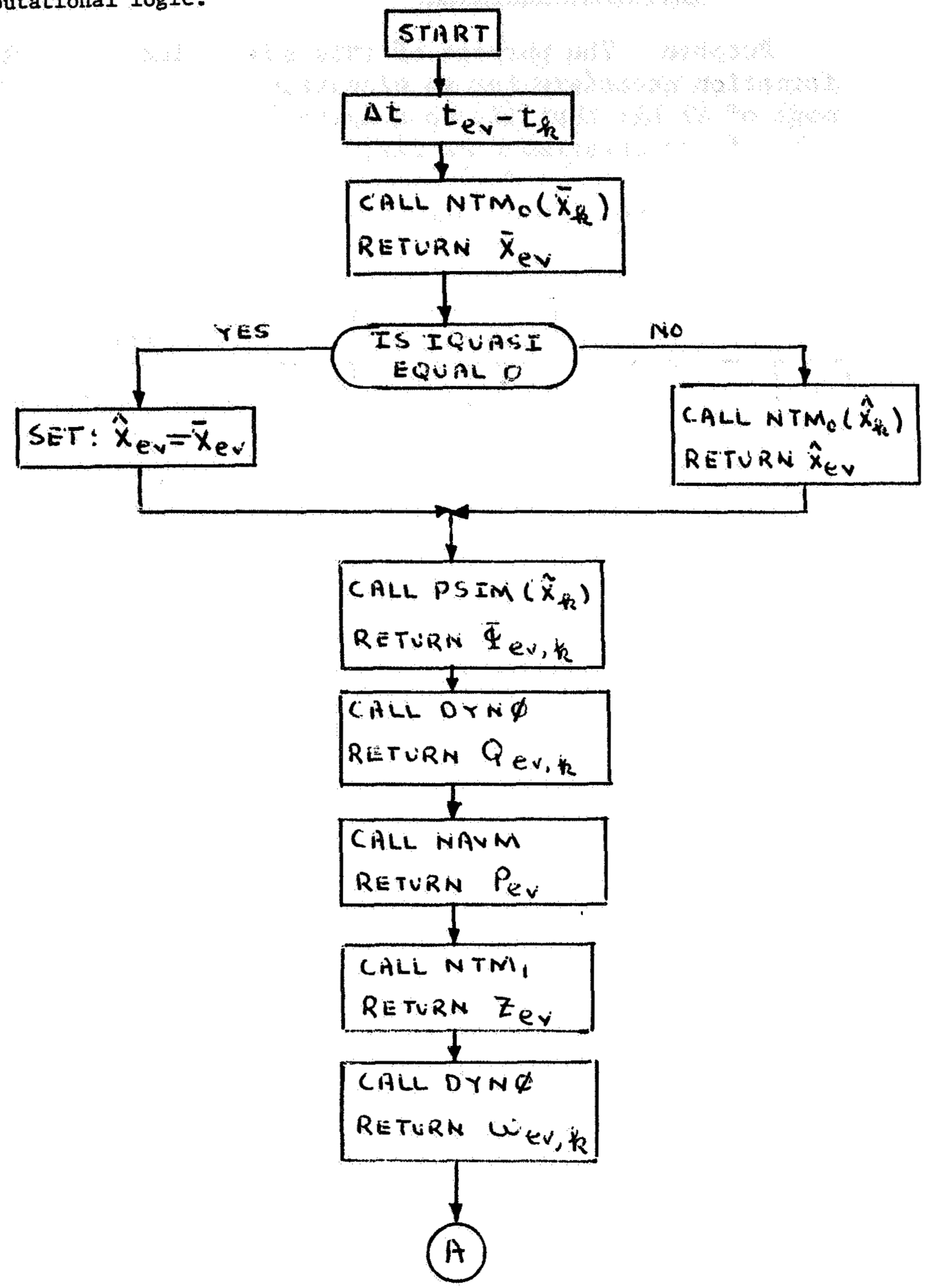
I/0	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Position and velocity components of the original nominal state vector
I	TEVN	t_{ev}	Time of the eigenvector event
I	RI1(6)	$\tilde{X}$	Position and velocity of the most recent nominal state vector.

Subprograms required: DYNO, HYELS, JACOBI, NAVM, NTM, PSIM.

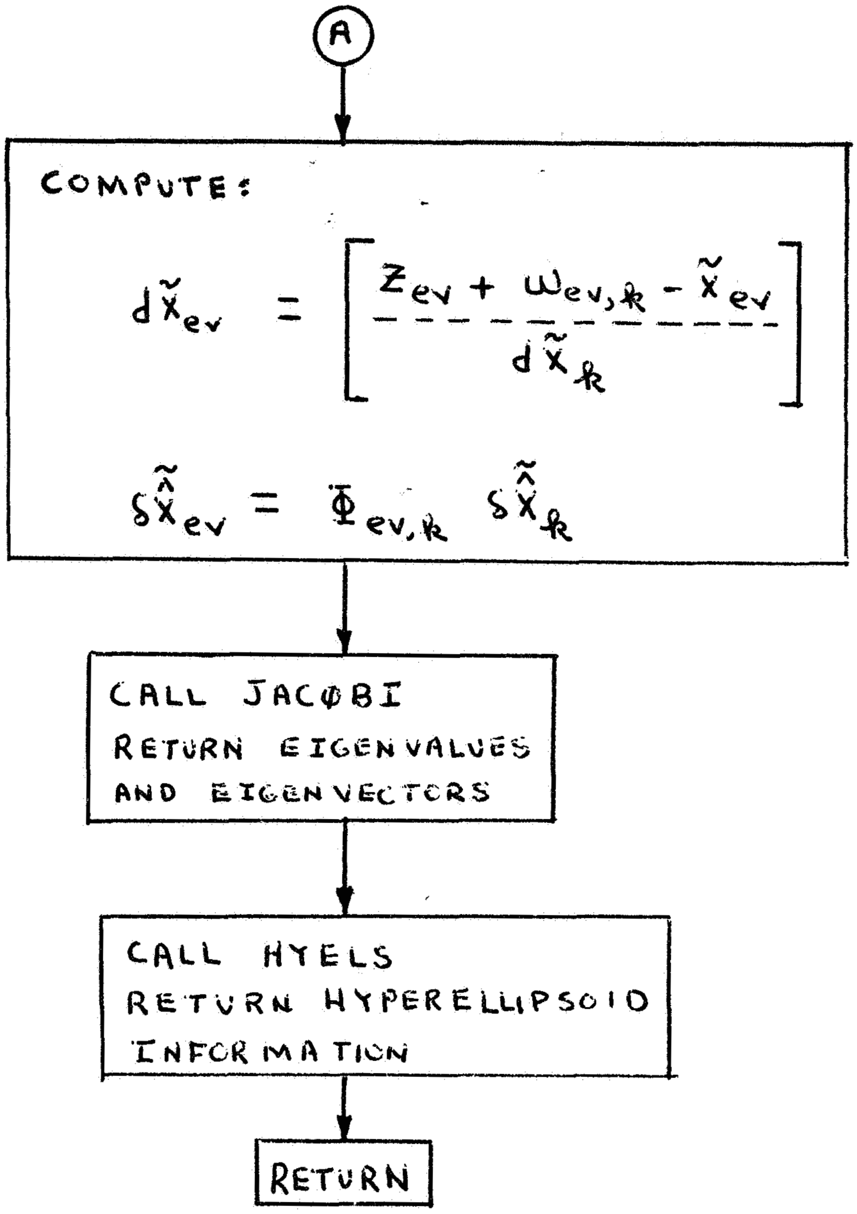
Approximate storage required (octal): 2620.

Discussion: As in the basic cycle the covariance matrix is propagated forward to the time of the eigenvector event. The eigenvalues and eigenvectors of that covariance matrix are then calculated together with the correlation coefficient matrix.

Computational logic:



Computational logic (concluded):



14. Subroutine EPHEM

Purpose: This subroutine computes the heliocentric ecliptic coordinates of a given planet at a specified time.

Calling Sequence: CALL EPHEM (N,D,ICODE).

Input/output:

I/O	Fortran name	Definition
I	N	Number of bodies for which the components will be found
I	D	Julian date, epoch 1900, at which the position and velocity will be computed.
I	ICODE	Internal code that states where to place the computed values of position and velocity (explained in greater detail in the discussion).

Subprograms required: None.

Approximate storage required (octal): 1260.

Discussion: In calculating the inertial coordinates of the planets, EPHEM makes use of the orbital elements that have been previously calculated in ORB: semimajor axis, eccentricity, inclination, longitude of ascending node, and longitude of perihelion. This subroutine then computes the mean anomaly of the planet and finally, the coordinates of the planet. While N specifies the number of planets for which the coordinates are to be computed, the vector NO contains the codes of the planets.

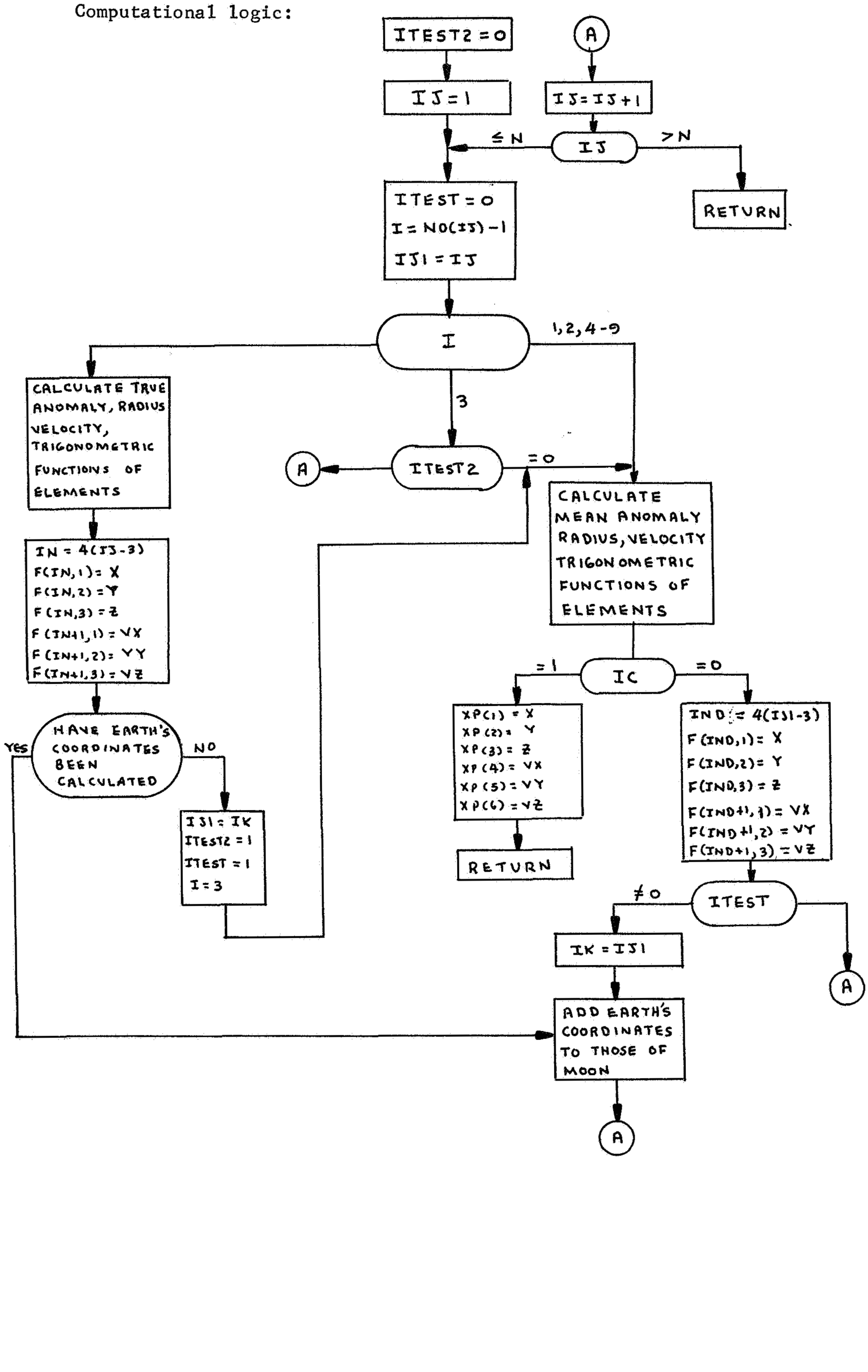
The subroutine allows two options:

1) If the coordinates for a specified planet are desired independent of the virtual mass program, ICODE should be set to one (1). The coordinate will then be placed in a vector XP;

2) For the virtual mass program EPHEM calculates the coordinates of all planets being considered in the analysis (N=NBODYI) and the coordinates are placed in an array F. For this option, ICODE = 0.

When either of the above options is exercised the units of the position and velocity returned are A.U. and A.U./day respectively.

Computational logic:



15. <u>Subroutine ESTMT</u>

Purpose: This subroutine acts as an auxiliary routine for VMP in computing the trajectory from the virtual mass technique. ESTMT updates the final values of the preceding computing interval to serve as initial values for the new step, determines the desired size of the next time increment, and estimates the final position and magnitude of the virtual mass.

Calling sequence: CALL ESTMT (DI, DELTM, TRTM)

Input/output:

I/O	Fortran name	Math symbol	Definition
I	DI		Julian date, epoch 1900, of the initial trajectory time
I	DELTM	Δt	Time interval over which the trajectory will be computed
I	TRTM	t_o	Initial trajectory time

Subprograms required: None.

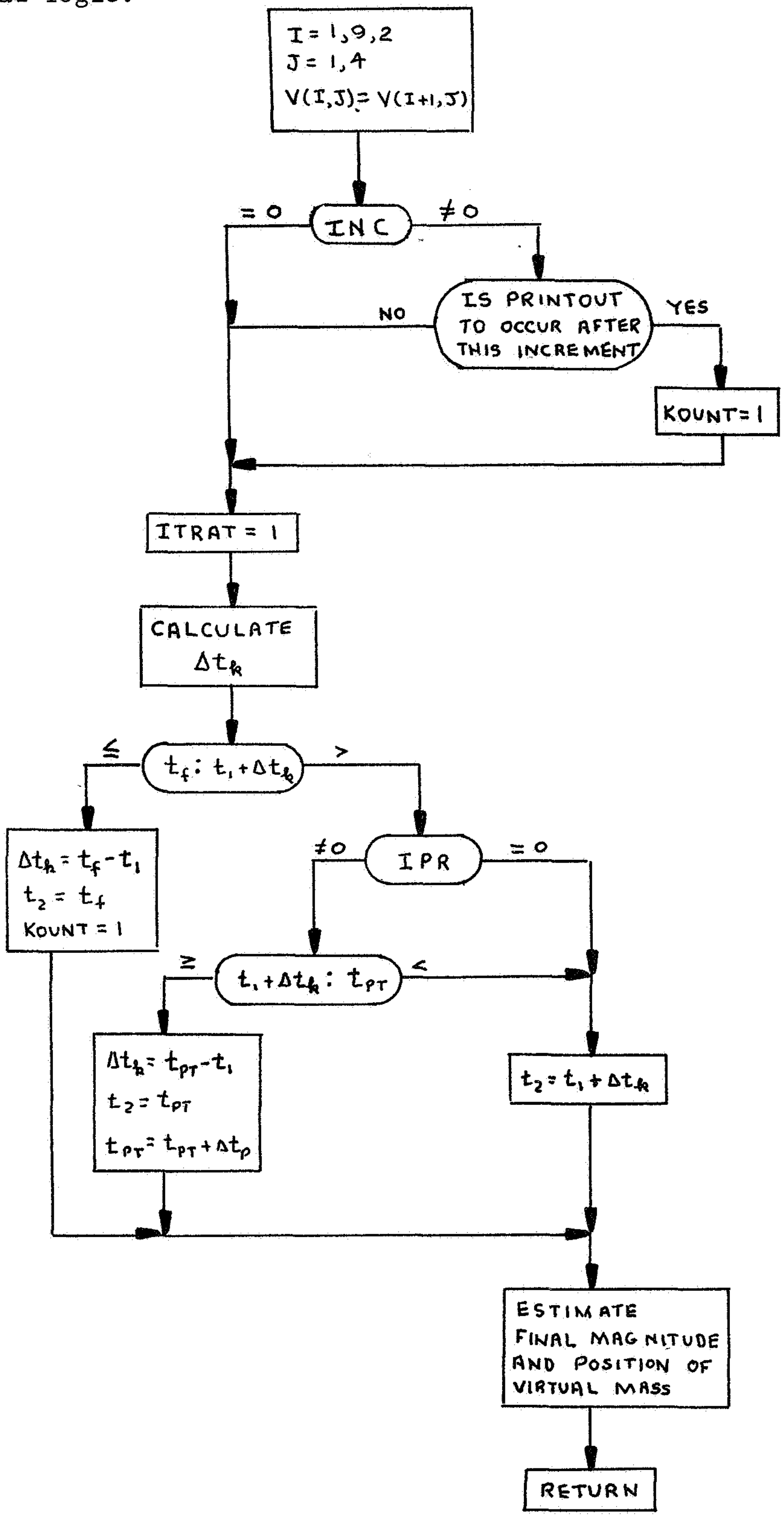
Approximate storage required (octal): 200.

Discussion: In determining the time increment to be used in the next step, ESTMT uses as a basis either the true anomaly increment or the requested print time.

It is necessary for the purposes of the nominal trajectory module to return the exact position and velocity of the vehicle at the end of the time period over which the trajectory is being computed. Therefore the final time itself must be computed as accurately as possible. Thus, $t_f = t_o + \Delta t$ and the ephemeris at the final time is based on this value of t_f, rather than $t_f = t_o + \sum_{k=1}^{n} \Delta t_k$ where Δt_k is the length of the k^{th} time increment computed by ESTMT.

For the formulas used in the subroutine to estimate the final position and magnitude of the virtual mass refer to Volume II the analytical manual of the final report.

Computational logic:



16. Subroutine EULMX

Purpose: This program computes the matrix required to define transformations from one coordinate system to another.

Calling sequence: CALL EULMX (ALP, NN, BET, MM, GAM, LL, P).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	ALP	α	First rotation angle
I	BET	β	Second rotation angle
I	GAM	γ	Third rotation angle
I	NN		First axis of rotation
I	MM		Second axis of rotation
I	LL		Third axis of rotation
O	P(3,3)		Transformation matrix

Subprograms required: None.

Approximate storage required (octal): 500.

Discussion: The program is a standard one computing the matrix that defines the transformation from one coordinate system to a new coordinate system obtained by rotating through an angle α about the first specified axis, β about the second, and γ about the third.

17. Subroutine GHA

Purpose: This routine computes the Greenwich hour angle and the universal time (in days) which is used in the tracking module to orient the tracking stations on a spherical rotating Earth.

Calling sequence: CALL GHA.

Input/output: All communication with this routine is accomplished through the use of common statements, which explains the lack of arguments.

Subprograms required: None.

Approximate storage required (octal): 70.

Discussion: In computing the Greenwich hour angle of the vernal equinox at some epoch T, the following equation is used:

$$GHA(T^*) = 100.0755426 + 0.985647346\ d + 2.9015 \times 10^{-13}\ d^2 + \omega t$$

for

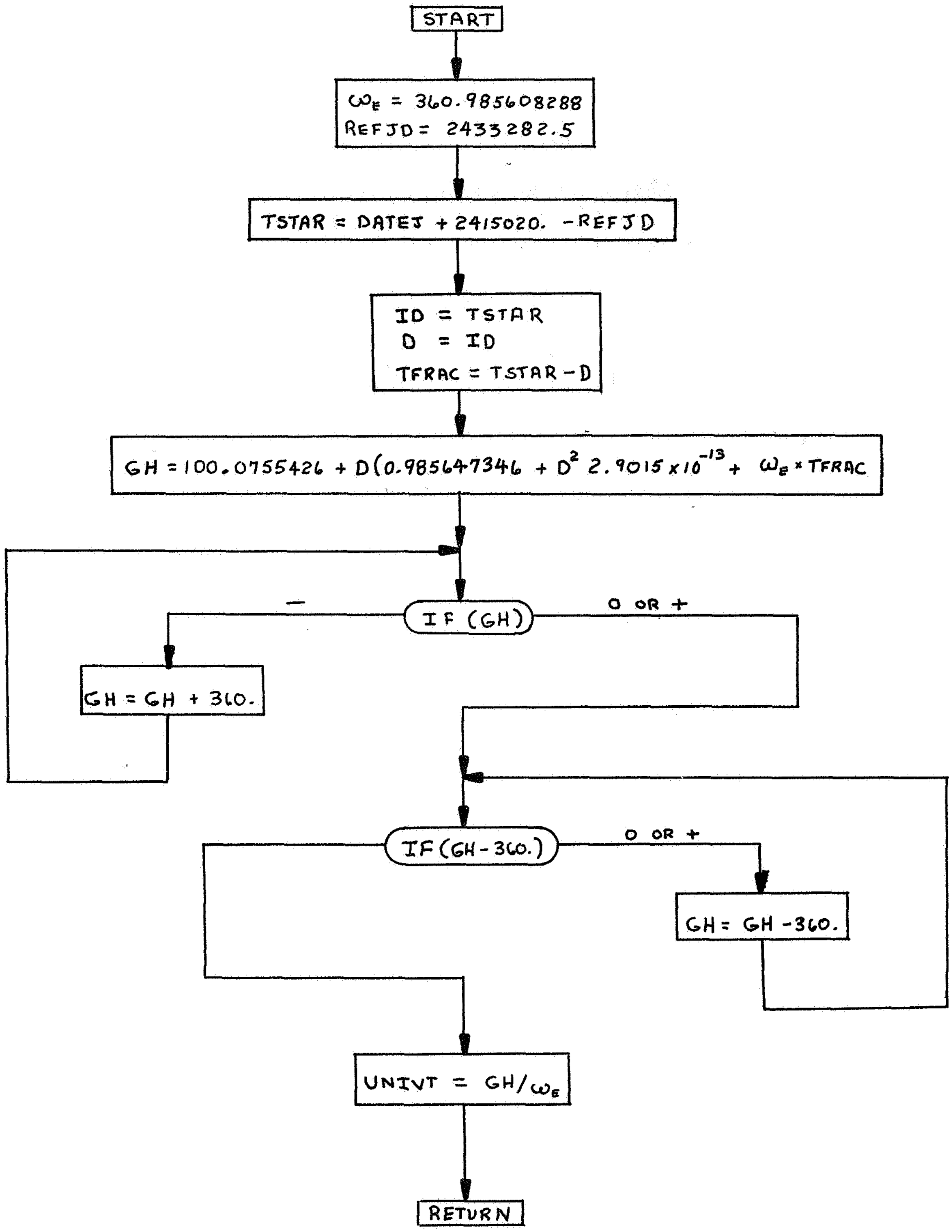
$$0 \leq GHA(T^*) < 360°$$

where d is the integer or whole days as determined by T^*, t is the fractional part of a day in seconds as determined by T^*, and ω is the Earth's rotation rate in degrees/day and is assumed constant.

The universal time is then computed in days from:

$$T = -\frac{GHA(T^*)}{\omega}$$

Computational logic:



18. Subroutine GUID

Purpose: In this subroutine Γ, the guidance matrix is computed, which is then returned to GUIDM to be used in computing the execution error matrix for a guidance event in the error analysis mode of STEAP.

Calling sequence: CALL GUID (RF, IGP, TEVN, GA, ADA).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RF(6)	$\bar{X}$	Position and velocity of the vehicle at the time of the guidance event
I	IGP		Guidance code describing which of three types of guidance policies is being used
I	TEVN	t_{ev}	Trajectory time of the guidance event
0	GA(3,6)	Γ	Guidance matrix
0	ADA(3,6)	η	Variation matrix

Subprograms required: EPHEM, HYELS, JACOBI, MATIN, NTM, ORB, PARTL, PSIM, VARADA.

Approximate storage required (octal): 3700.

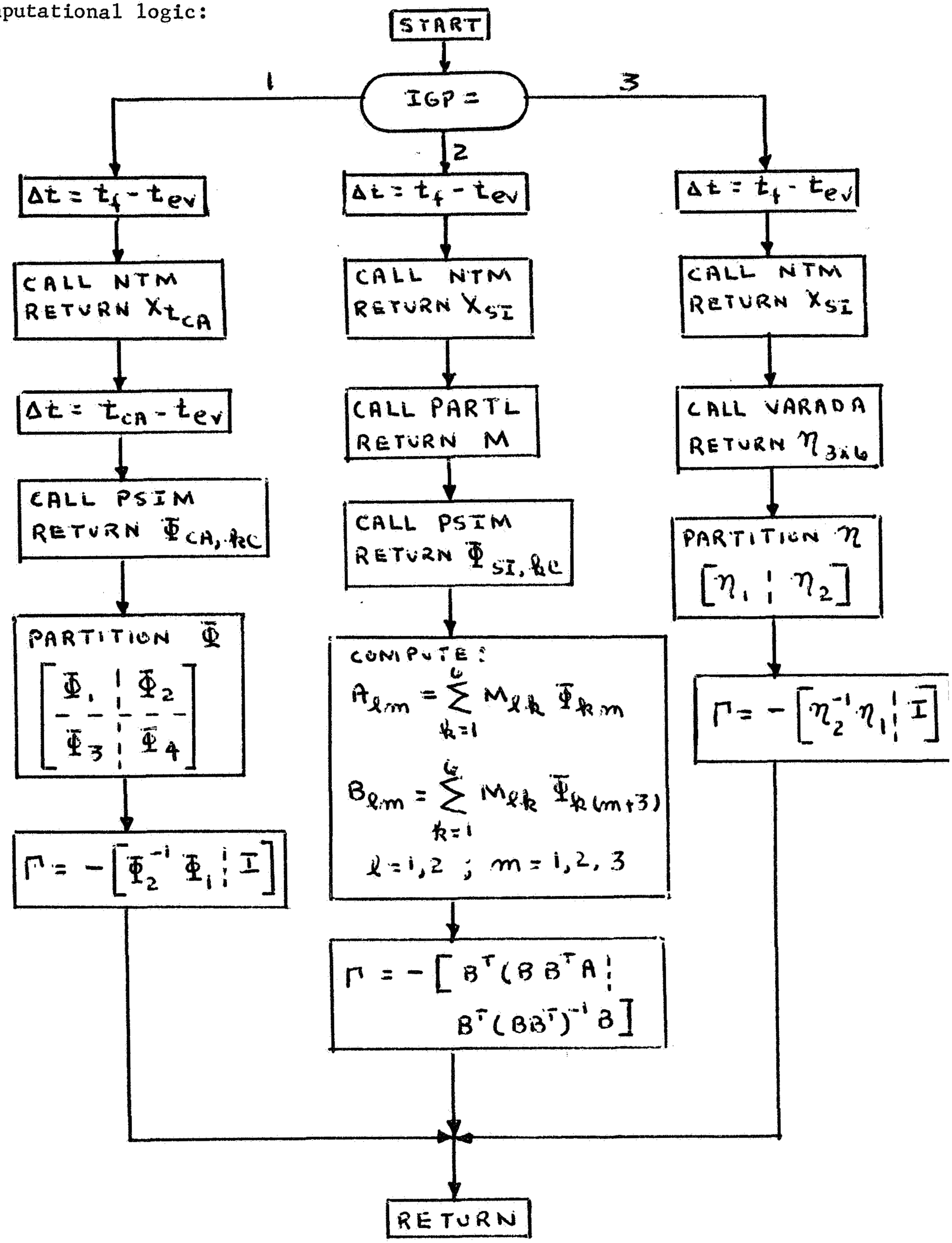
Discussion: The type of guidance policy is determined immediately. If fixed-time-of-arrival policy is being used, the conditions at closest approach are determined. The M matrix is obtained (assuming the vehicle has passed through the sphere of influence) to be used later in a prediction event. The state transition matrix is then obtained relating deviations at the time of the guidance event to those at closest approach. The guidance matrix, Γ, is computed, finally from the state transition matrix.

If either two-variable B-plane or three-variable B-plane guidance policy is to be used, the conditions at sphere of influence are obtained. Again, the M matrix is computed. Now, if two-variable B-plane guidance policy is desired, the state

transition matrix relating deviations at the time of the guidance policy to those at sphere of influence are obtained. Then guidance submatrices A and B are calculated, from which the guidance matrix is computed. For a further discussion see Volume II.

Finally, if three-variable B-plane guidance policy is used the variation matrix, η, is constructed from VARADA, which is used to compute the guidance matrix.

Computational logic:



19. Subroutine GUIDM

Purpose: The execution error matrix used in the guidance event in the error analysis mode of STEAP is computed in this subroutine. In addition, other pertinent information is calculated and printed.

Calling sequence: CALL GUIDM (RI, TEVN).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}$	Position and velocity of the vehicle at the time of the last measurement or event
I	TEVN	t_{ev}	Trajectory time of the guidance event

Subprograms required: DYNO, GUID, HYELS, JACOBI, NAVM, NTM, PSIM.

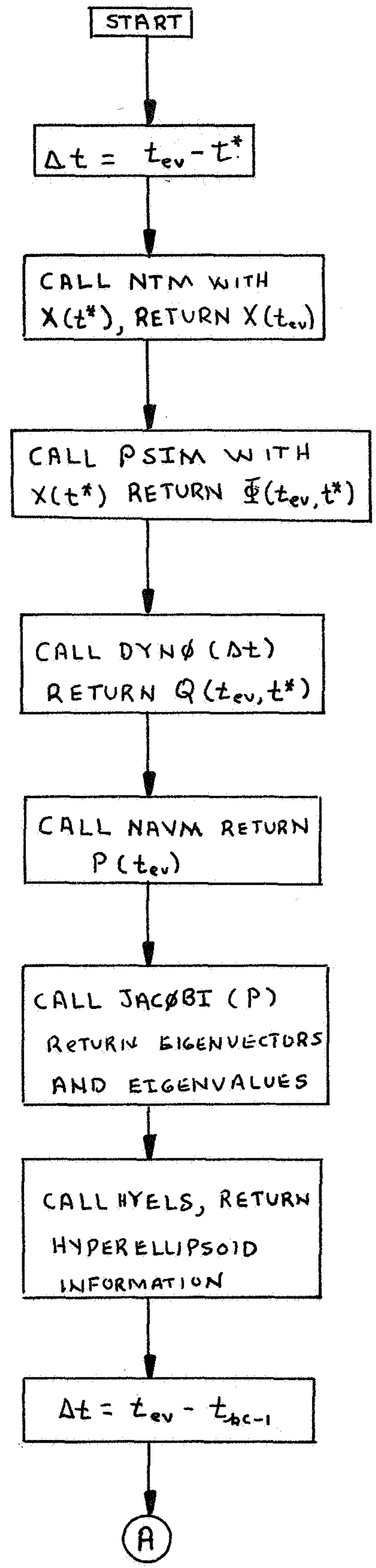
Approximate storage required (octal): 5360.

Discussion: This subroutine is responsible for all the logic at a guidance event for the error analysis mode. In general it determines the covariance matrix prior to the correction, calls the GUID subroutine for the guidance matrix depending on the guidance policy, and computes the covariance matrix associated with the velocity components.

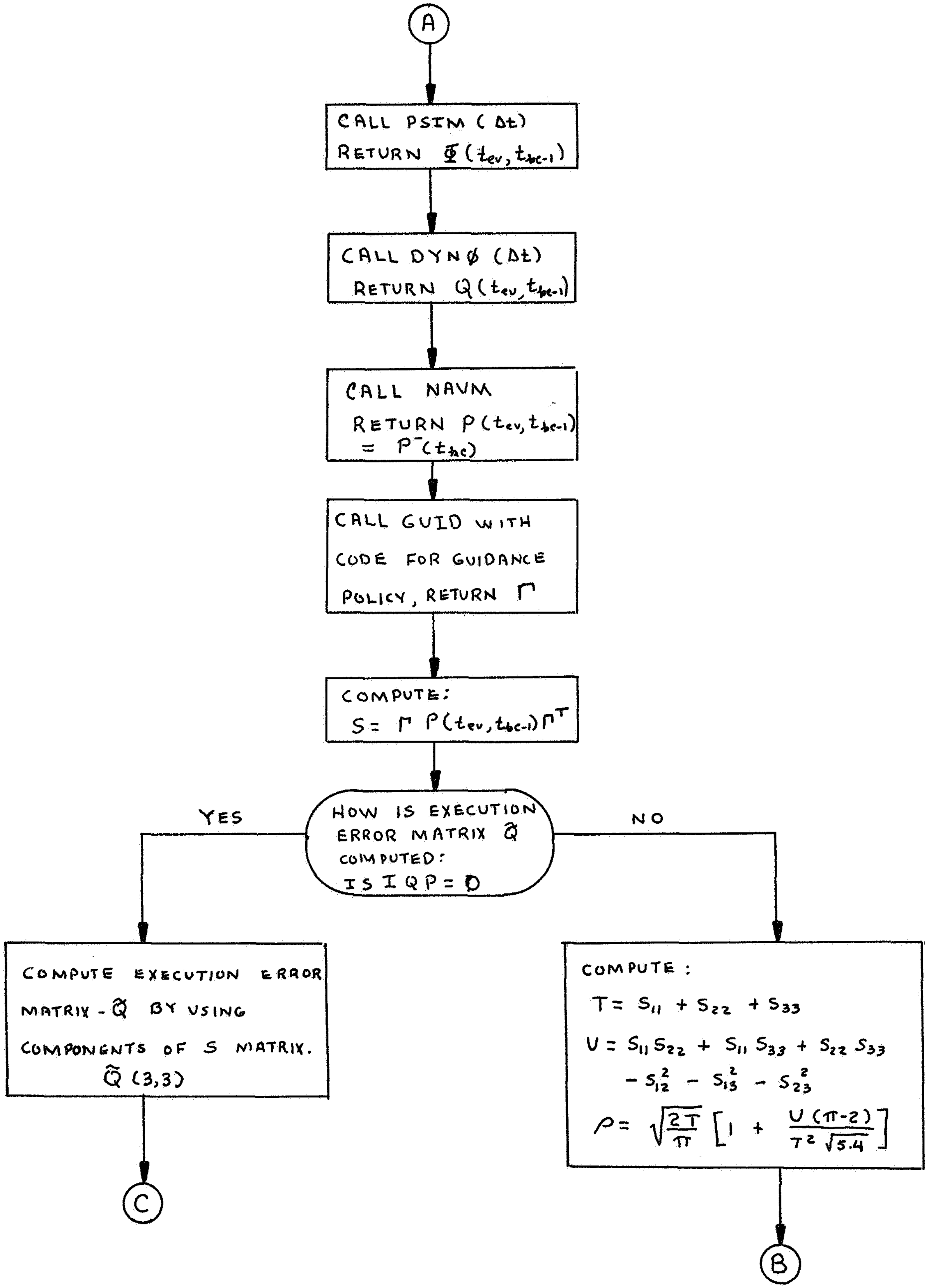
In addition, GUIDM computes the execution error matrix by one of two methods as determined by input. The final output of this subroutine is the covariance matrix of errors if a correction would have been made.

Other computations are made throughout the logic to determine eigenvector, eigenvalues, and hyperellipsoid information of various matrices.

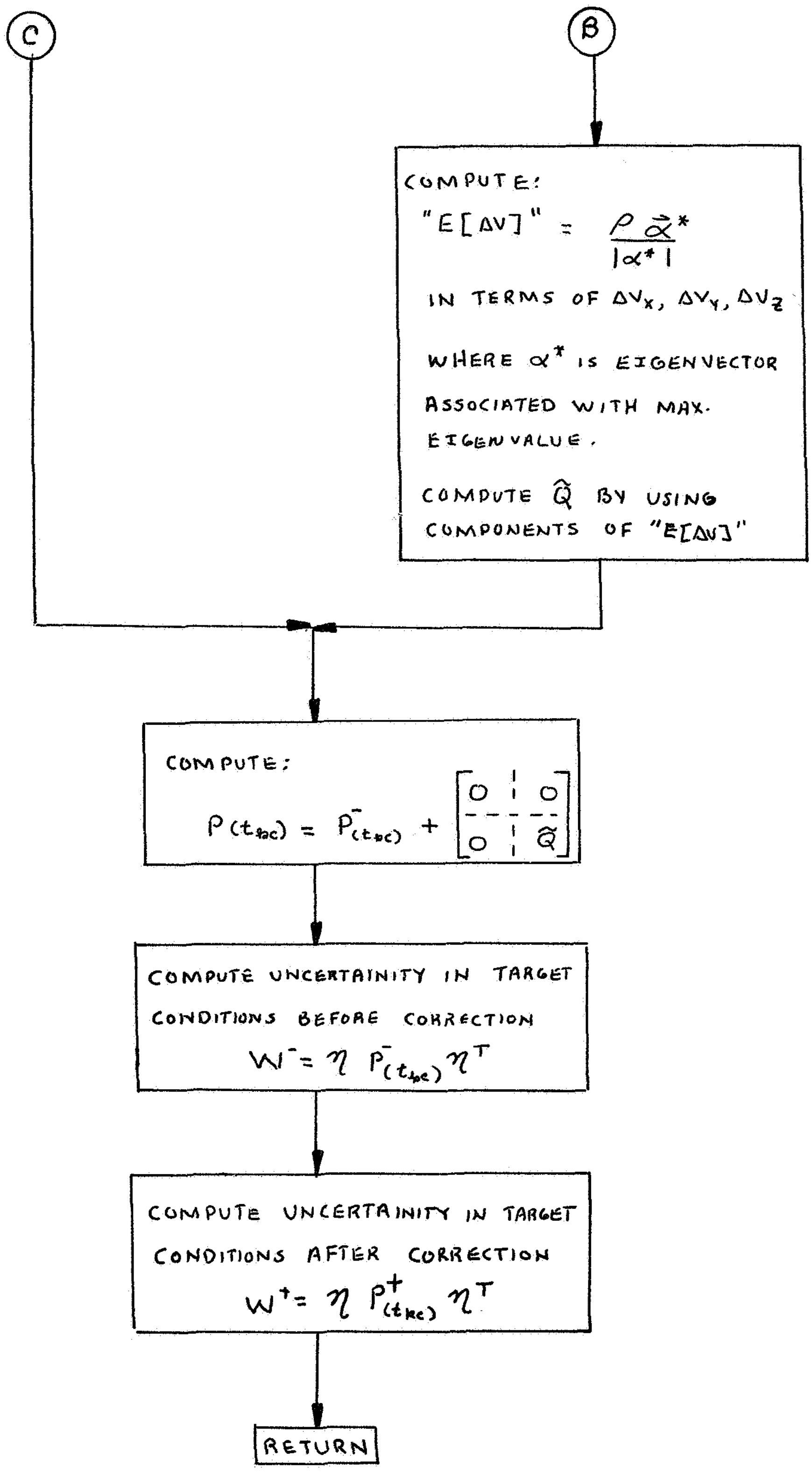
Computational logic:



Computational logic (continued):



Computational logic (concluded):



20. Subroutine GUIS

Purpose: This subroutine computes Γ, the guidance matrix, for use in the guidance event for the simulation mode.

Calling sequence: CALL GUIS (RF, RF1, IGP, TEVN, GA, ADA).

Input/output:

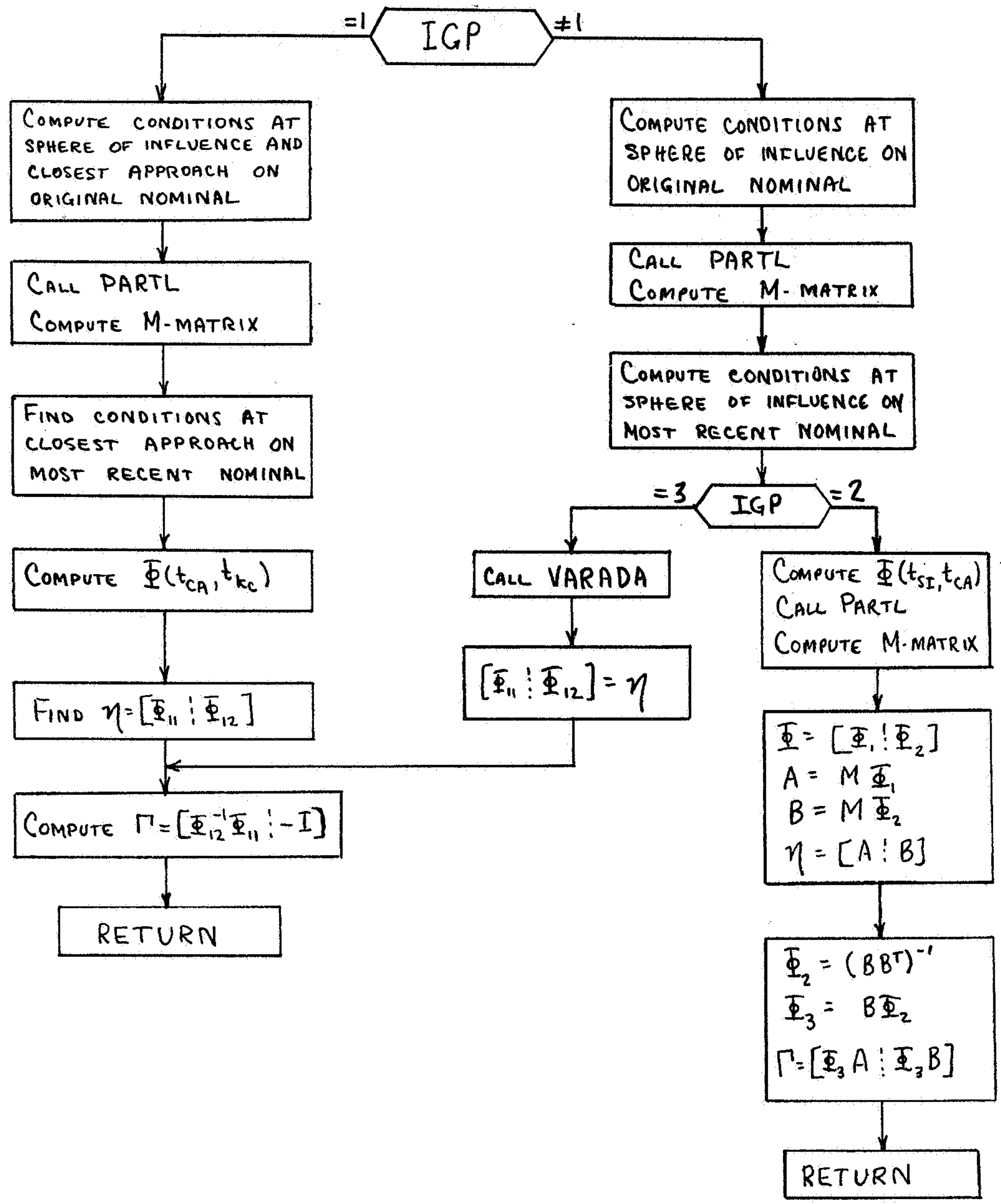
I/O	Fortran name	Math symbol	Definition
I	RF(6)	$\overline{X}$	Position and velocity of the vehicle on the original nominal trajectory at the time of the guidance event
I	RF1(6)	$\tilde{X}$	Position and velocity of the vehicle on the most recent nominal trajectory at the time of the guidance event
I	IGP		An internal code which determines which type of guidance policy is being used
I	TEVN	t_{ev}	Trajectory time of the guidance event
0	GA(3,6)	Γ	Guidance matrix
0	ADA(3,6)	η	Variation matrix

Subprograms required: EPHEM, HYELS, JACOBI, MATIN, NTM, ORB, PARTL, VARSIM.

Approximate storage required (octal): 4470.

Discussion: A similar method to that described for GUID is employed to produce the gamma matrix, Γ. However, a difference should be noted in the calculation of the M matrix. This is computed from the information at sphere of influence on the original nominal trajectory while the rest of the calculations are based on the most recent nominal trajectory.

Computational logic:



21. Subroutine GUISIM

Purpose: This subroutine is responsible for the logic contained in the guidance event of the simulation mode.

Calling sequence: CALL GUISIM (RI, TEVN, RI1).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Position and velocity of the vehicle on the original nominal trajectory at the time of the last measurement or event.
I	TEVN	t_{ev}	Trajectory time of the guidance event.
I	RI1(6)	$\tilde{X}$	State of the vehicle on the most recent nominal trajectory at the time of the guidance event.

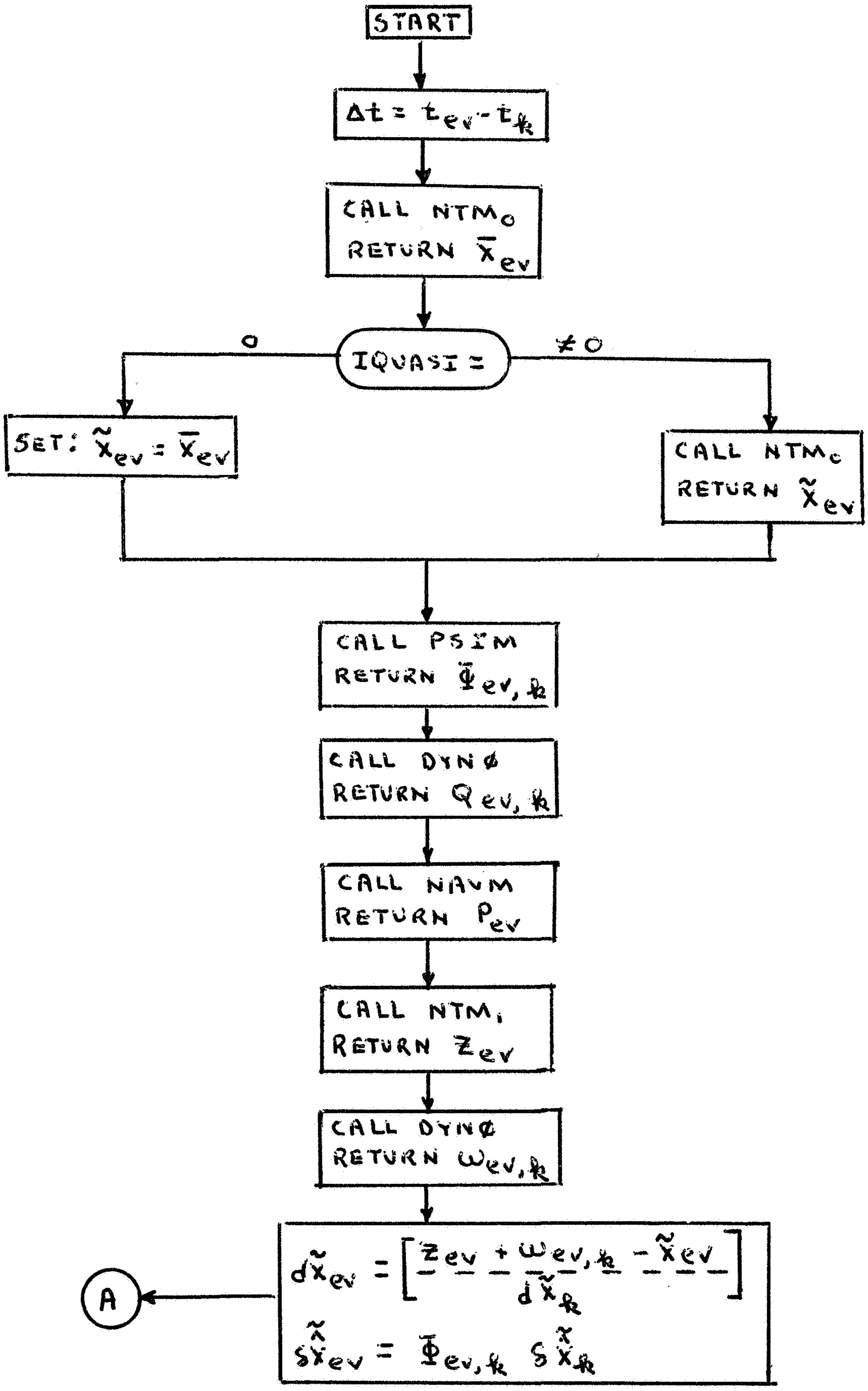
Subprograms required: DYNO, GUIS, HYELS, JACOBI, NAVM, NTM, PSIM.

Approximate storage required (octal): 5420.

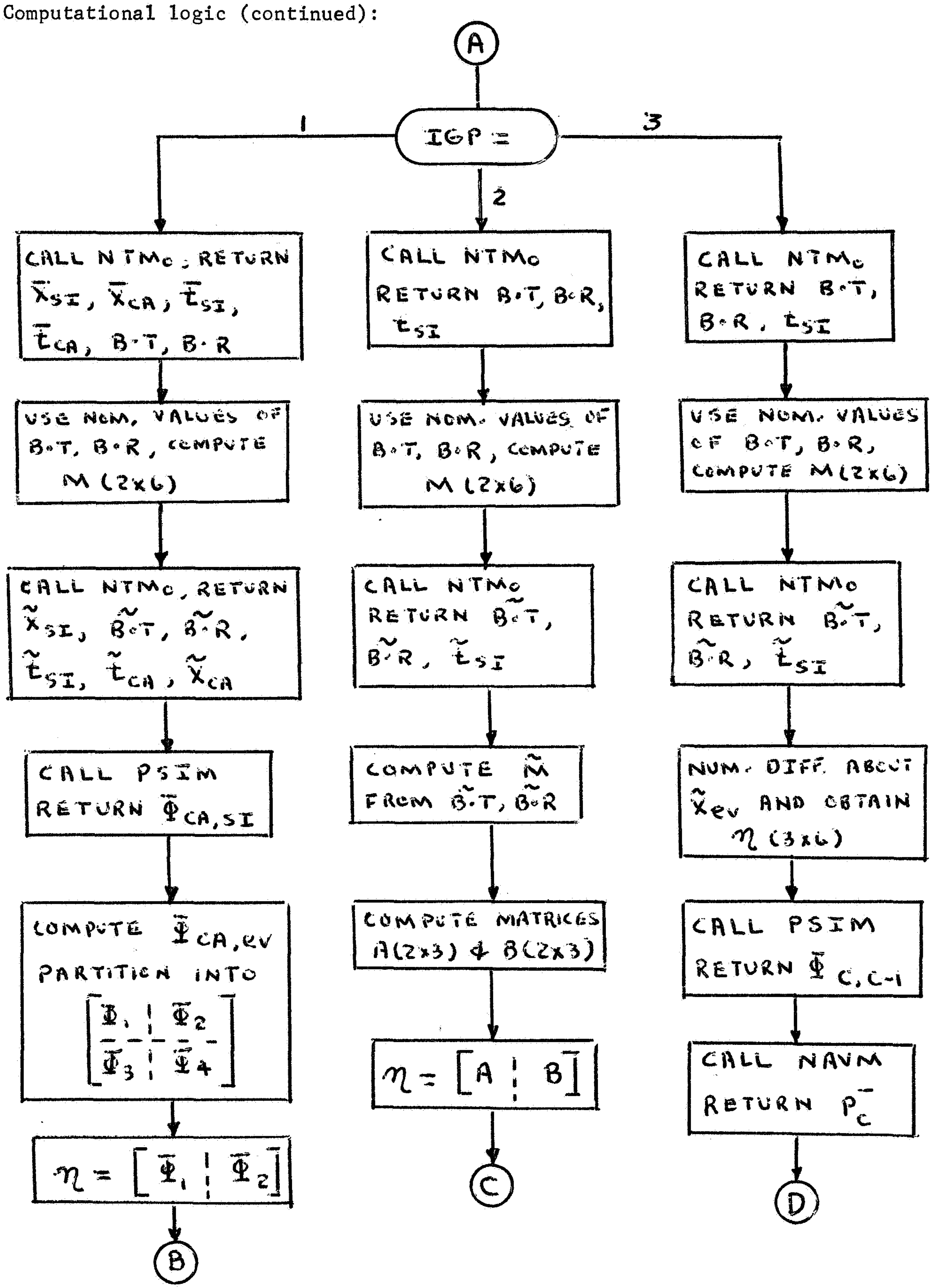
Discussion: The GUISIM subroutine is similar to the error analysis subroutine GUIDM in that it develops the logic at a guidance event for the simulation mode.

It computes the execution error matrix, the covariance matrix before and after a correction, and the probabilistic uncertainties in the target conditions before and after the correction. Additional computations are made to determine the commanded correction ΔV_c, the perfect correction $\underline{\Delta V}$, and the error in correction due to navigation uncertainty. The actual correction $\overline{\Delta V}$ is also determined.

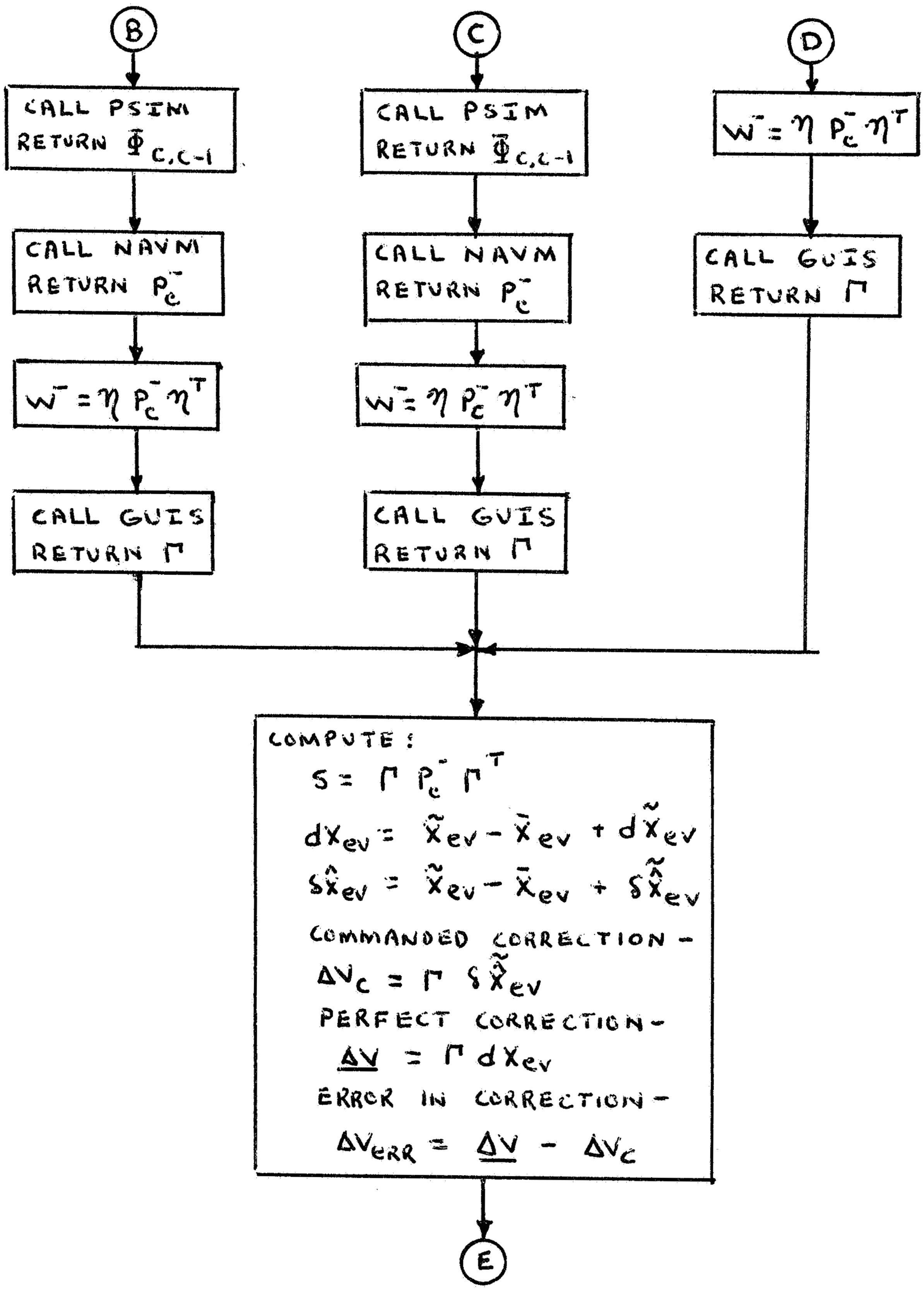
Computational logic:



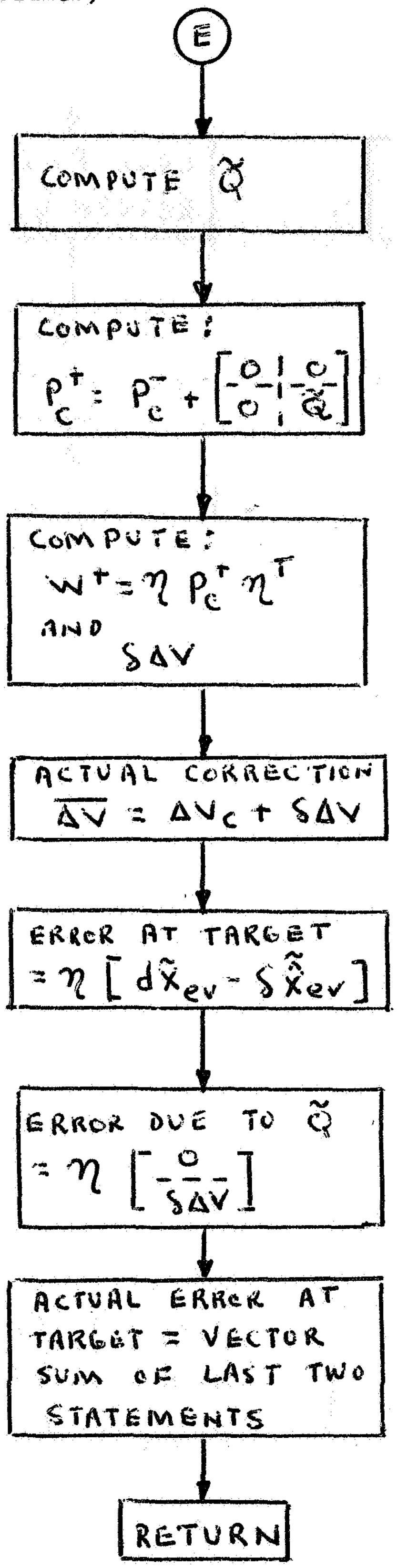
Computational logic (continued):



Computational logic (continued):



Computational logic (concluded):



22. Subroutine HYELS

Purpose: The two-dimensional or three-dimensional hyper-ellipsoid of a specified matrix is computed and printed.

Calling sequence: CALL HYELS (KS, P, N).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	KS	σ	Sigma level of the hyper-ellipsoid.
I	P(3,3)	P	Matrix for which the hyper-ellipsoid is to be computed.
I	N	n	Dimension of the square matrix P.

Subprograms required: MATIN.

Approximate storage required (octal): 410.

Discussion: The subroutine MATIN is used to compute the inverse of the matrix P which is a square matrix of dimension 2 x 2 or 3 x 3. The three-dimensional hyperellipsoid is then computed as

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz = \sigma^2$$

where

$$a = P^{-1}\ (1,1)$$

$$b = P^{-1}\ (2,2)$$

$$c = P^{-1}\ (3,3)$$

$$d = 2\left[P^{-1}\ (1,2)\right]$$

$$e = 2\left[P^{-1}\ (1,3)\right]$$

$$f = 2\left[P^{-1}\ (2,3)\right]$$

For the two-dimensional hyperellipsoids the appropriate component is set to zero.

23. Subroutine HYPER

Purpose: This program computes the elements of the launch hyperbola on which the injection conditions are based.

Calling sequence: CALL HYPER (S, RP, VHL, GME, ELAT, A, E, XI, XL, XW, W, PV, Q, AZ, C3, P, DLA, RAL).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	S(3)	$\hat{S}$	Hyperbolic excess velocity (equatorial coordinates)
I	RP	r_P	Periapsis radius (= 6560 km)
I	VHL	V_{HL}	Speed at infinity along hyperbola
I	GME	μ_P	Gravitational constant of launch planet
I	ELAT	φ_L	Latitude of launch site
0	A	a	Semimajor axis
0	E	e	Eccentricity
0	XI	i	Inclination
0	XL	Ω	Longitude of ascending node
0	XW	ω	Argument of periapsis
0	W(3)		Normal to plane
0	PV(3)	$\hat{P}$	Unit vector directed toward periapsis
0	Q(3)	$\hat{Q}$	Unit vector normal to P in orbital plane
I	AZ	Σ_L	Launch azimuth

0	C3	C_3	Launch energy
0	P	P	Semilatus rectum
0	DLA	δ	Declination of departure asymptote
0	RAL	Θ	Right ascension of departure asymptote

Subprograms required: None.

Approximate storage required (octal): 600.

Discussion: This program is adapted from programs in the SPARC program (ref 1). The nominal launch azimuth is set for a due east launch, but if that is impossible (with the approaxh asymptote constraints) it is reset by the program to a realistic value. The periapsis radius is set equal to the desired parking orbit radius. Otherwise the program is a standard conic program.

24. Subroutine HYPSV

Purpose: This program computes the position and velocity vectors in ecliptic and equatorial coordinates and time from periapsis at a given radius on a specified hyperbola.

Calling sequence: CALL HYPSV (R, P, E, C3, VHL, GME, RP, PV, Q, TA, XEQ, VEQ, VS, GAM, TS, XEC, VEC, ECEQ).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	R	r	Radius at which state is desired
I	P	P	Semilatus rectum
I	E	e	Eccentricity
I	C3	C_3	Energy
I	VHL	V_{HL}	Hyperbolic excess velocity
I	GME	μ_P	Gravitational constant of primary
I	RP		Periapsis radius
I	PV(3)	$\hat{P}$	Unit vector to periapsis from primary
I	Q(3)	$\hat{Q}$	Unit vector normal to P in plane or orbit
O	TA	ν	True anomaly at given radiu
O	XEQ(3)	X_{eq}	Position vector at given radius (equatorial coordinates)
O	VEQ(3)	V_{eq}	Velocity vector at given radius (equatorial coordinates)
O	VS		Speed at given radius
O	GAM	γ	Path angle at given radius

0	TS		Time (sec) from periapsis to radius
0	XEC(3)	X_{EC}	Position vector at given radius (ecliptic coordinates)
0	VEC(3)	V_{EC}	Velocity vector at given radius (ecliptic coordinates)
I	ECEQ(3)	M_{ECEQ}	Transformation matrix from ecliptic to equatorial coordinates

Subprograms required: none.

Approximate storage required (octal): 500.

Discussion: This program is a standard conic program.

25. Subroutine INPUTZ

Purpose: This subroutine is responsible for converting the input information for the virtual mass program into variables compatible with the rest of the virtual mass subroutines.

Calling sequence: CALL INPUTZ (RS` NTP` IPRINT).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RS(6)	$\overline{X}$	Heliocentric ecliptic coordinates of the vehicle at the initial time.
I	NTP		Code number of the target planet.
I	IPRINT		An internal code used to determine if the printing of initial information is desired.

Subprograms required: NEWPGE, SPACE, TIME.

Approximate storage required (octal): 520.

Discussion: This routine converts all input information into the proper units in addition to setting the correct variable names and printing the initial information if desired. If the input variable IPRINT = 0, the initial data will be printed. Otherwise, no initial printout will occur.

26. Subroutine JACOBI

Purpose: The eigenvalues and eigenvectors of a given matrix are computed and returned.

Calling sequence: CALL JACOBI (A, W2, V, N, FOD).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	A(1)	A	Input matrix to be diagonalized (will be destroyed).
0	W2(1)		Output vector of eigenvalues.
0	V(1)		Output matrix of eigenvectors [size (N, N)].
I	N	n	Dimension of square matrix A.
I	FOD		Final off-diagonal annihilation value.

Subprograms required: None.

Approximate storage required (octal): 440.

Discussion: The subroutine uses the threshold version of the Jacobi method for computing eigenvalues and eigenvectors of A. The A matrix should be real and symmetric.

27. Subroutine LAMB

Purpose: This program solves for heliocentric ecliptical transfer orbits that are specified by an initial radius, a final radius, a central angle, and a time of flight.

Calling sequence: CALL LAMB (RL, RP, PSI, TF, GM, LOC, NTYS, A, E, P, VL, VP).

Input/output:

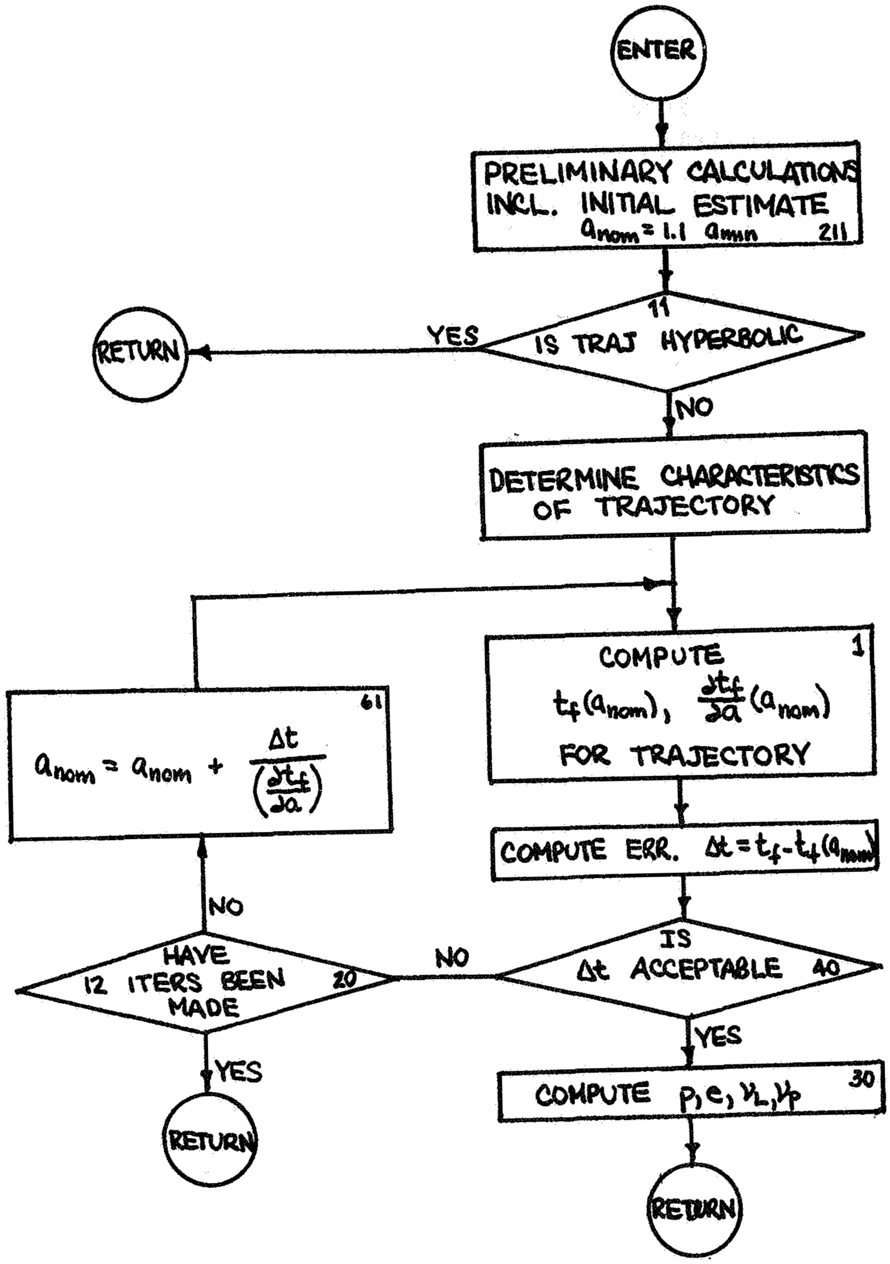
I/0	Fortran name	Math symbol	Definition
I	RL	R_L	Heliocentric launch planet radius
I	RP	R_P	Heliocentric target planet radius
I	PSI	Ψ	Transfer angle, radi
I	TF	t_f	Time of flight, days
I	GM	μ	Gravitational constant of sun
0	LOC		Flag indicating whether iterative process converged (LOC 4) or failed (LOC 5).
I	NTYS		NTYS = 1 for $0 \leq \psi \leq 180°$ = 2 for $180° \leq \psi \leq 360°$
0	A	a	Semimajor axis of heliocentric ellipse
0	E	e	Eccentricity of heliocentric ellipse
0	P	P	Semilatus rectum of heliocentric ellipse
0	VL	ν_L	True anomaly at launch, radi
0	VP	ν_P	True anomaly at arrival, radi

Subprograms required: None.

Approximate storage required (octal): 1100.

Discussion: This program is a simplified version of the Lambert-theorem program LAMC discussed in reference 1. LAMB does not compute hyperbolic cases; however, since energy limitations do not allow heliocentric hyperbolic transfers from Earth launch this does not seem a severe restriction.

Computational logic:



28. Subroutine MATIN

Purpose: This subroutine computes the inverse of the input matrix.

Calling sequence: CALL MATIN (A, R, N).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	A(1)	A	Matrix that is to be inverted.
O	R(1)	R	Inverse of matrix A.
I	N	n	Size of matrix A.

Subprograms required: None.

Approximate storage required (octal): 1610.

Discussion: The subroutine uses the bordering method of matrix inversion. Matrices A and R may share the same locations in which case A is destroyed.

29. Subroutine MENO

Purpose: The measurement noise matrix is determined and returned to the basic cycle. Alternately, the actual measurement noise may be determined.

Calling sequence: CALL MENO (MMCODE, ICODE).

Input/output:

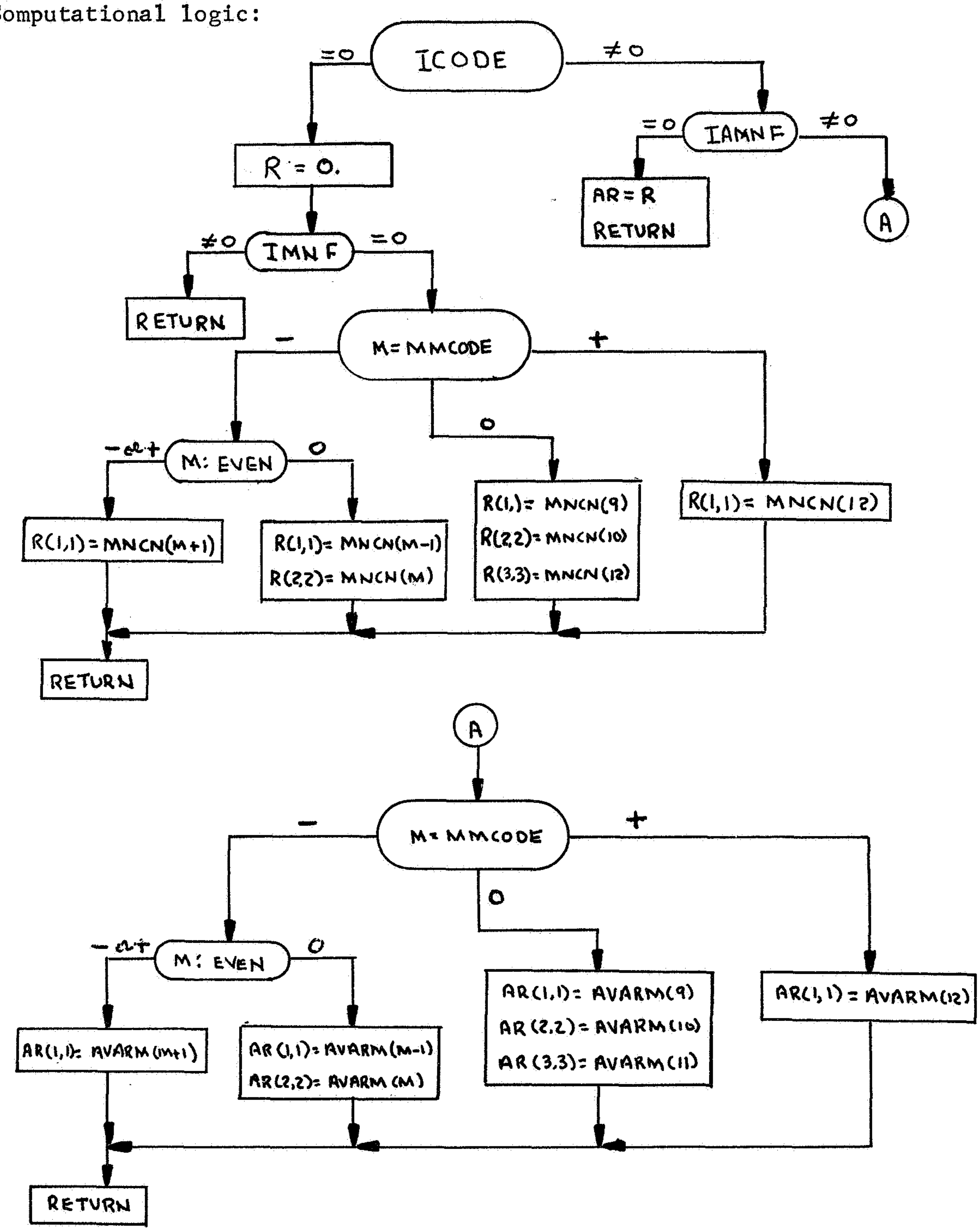
I/O	Fortran name	Definition
I	MMCODE	Measurement model code.
I	ICODE	Internal code used to distinguish between the two alternative listed above.

Subprograms required: None.

Approximate storage required (octal): 140.

Discussion: The measurement noise is input for each type of measurement. MENO chooses the correct value according to MMCODE and places it in the appropriate location. If ICODE = 0, MENO computes the measurement noise matrix, R, for both the error analysis and simulation modes. However, if the actual measurement noise matrix, AR, is desired for the simulation mode, ICODE = 1.

Computational logic:



30. Subroutine MUND

Purpose: MUND is responsible for computing the augmented portion of the state transition matrix when the gravitational constant of the Sun or of the target planet has been augmented to the basic state vector.

Calling sequence: CALL MUND (RI, RF, POSS).

Input/output:

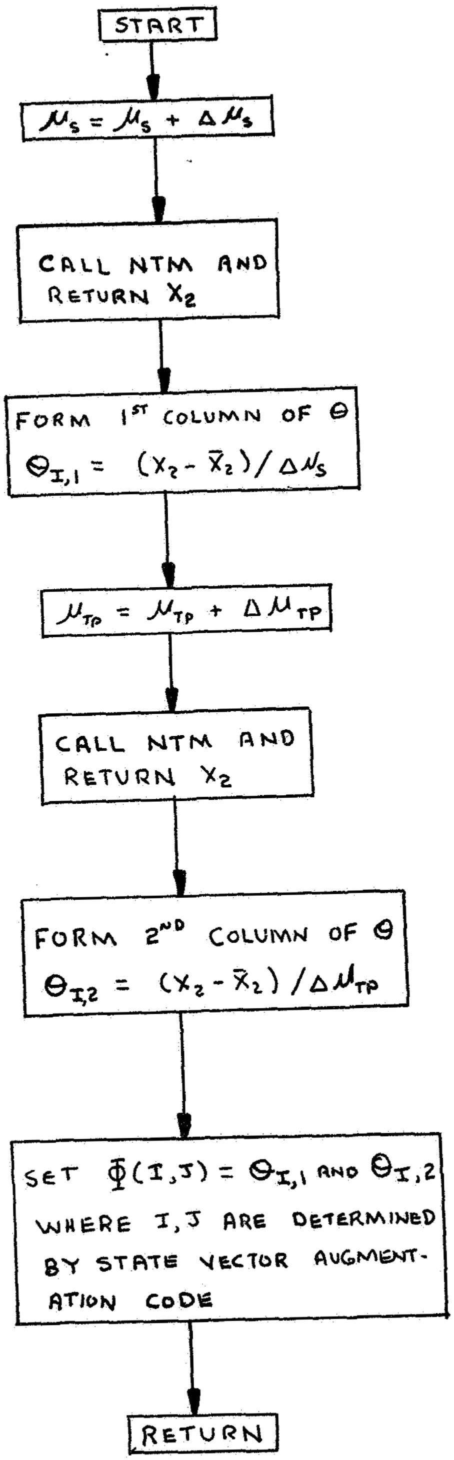
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}_i$	Position and velocity of the vehicle at the beginning of the time interval.
I	RF(6)	$\overline{X}_f$	Position and velocity of the vehicle at the end of the time interval.
I	POSS	r	Distance of the vehicle from the target planet at the initial time.

Subprograms required: NTM.

Approximate storage required (octal): 240.

Discussion: A numerical differencing technique is used to compute that augmented portion of the state transition matrix which relates to the gravitational constants of the Sun and of the target planet. The amount by which the gravitational constant of either body is altered may be input as data or in their absence, the program will assume the values specified in DATA. The portion of the state transition matrix relating deviations of the gravitational constant of the target planet will be assumed zero until the vehicle approaches a distance from the target planet of six times the sphere of influence of the planet.

Computational logic:



31. Subroutine NAVM

Purpose: The navigation module propagates the covariance matrix from the time of the last measurement or event to the present through the use of the standard Kalman algorithm.

Calling sequence: CALL NAVM (NR, ICODE).

Input/output:

I/O	Fortran name	Definition
I	NR	Number of rows in the measurement noise matrix.
I	ICODE	Internal code which determines if a measurement is being processed (see dicussion).

Subprograms required: MATIN.

Approximate storage required (octal): 470.

Discussion: The standard Kalman filtering equations are used to propagate the covariance matrix P.

$$P_k^- = P\left(t_k, t_{k-1}\right) = \Phi_{k,\ k-1}\ P_{k-1}^+\ \Phi_{k,\ k-1}^T + Q_{k,\ k-1}$$

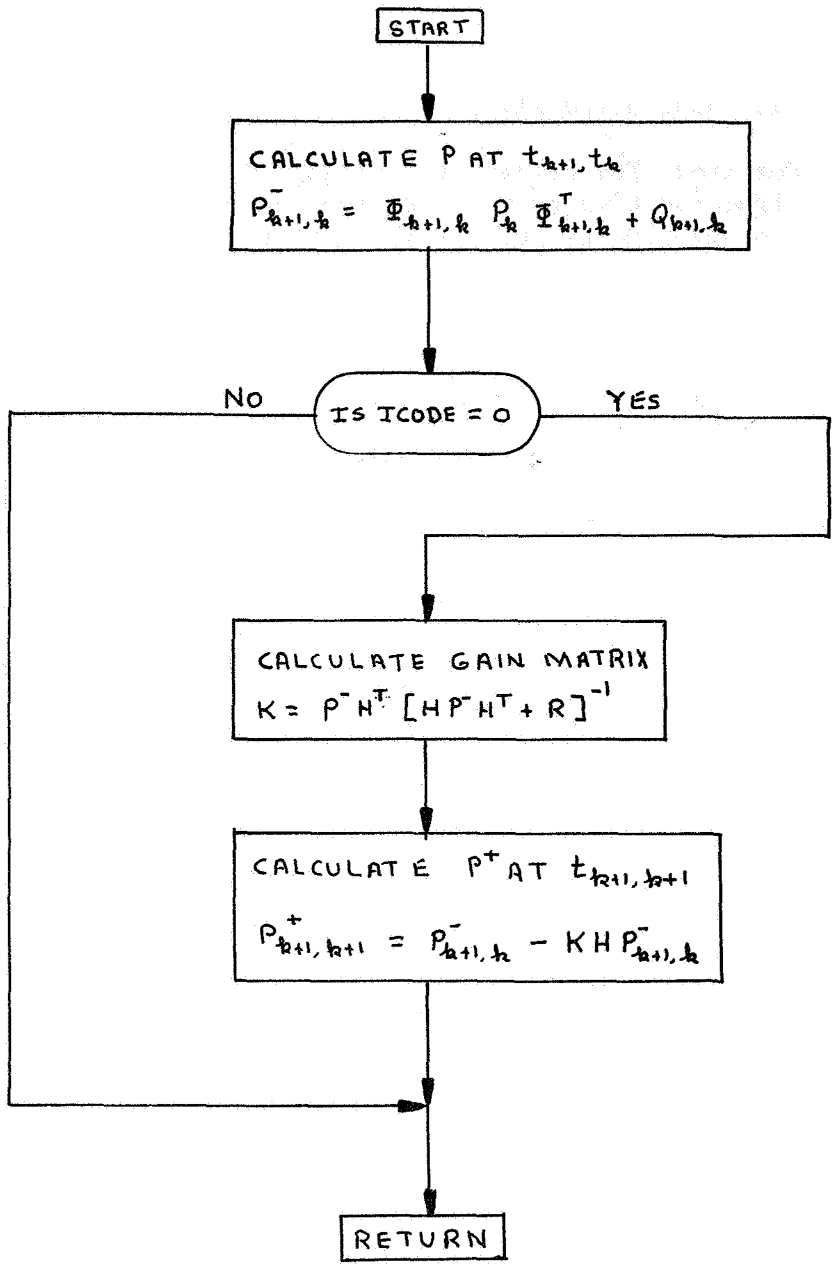
If ICODE = 1, no measurement is being processed and $P_k^- = P_k^+$ is returned as the covariance matrix at time t_k. However, if a measurement is indeed being processed, ICODE = 0 and P_k^+ is calculated.

$$K_k = P_k^-\ H_k^T \left(H_k\ P_k^-\ H_k^T + R_k\right)^{-1}$$

also $P_k^+ = P_k^- - K_k\ H_k\ P_k^-$

Note: If no measurement is being processed (as in an event) the calling sequence should be CALL NAVM (1, 1) as the value of NR is unimportant, but must appear.

Computational logic:



32. Subroutine NDTM

Purpose: The numerical differencing technique is used to compute the unaugmented portion of the state transition matrix.

Calling Sequence: CALL NDTM (RI, RF).

Input/output:

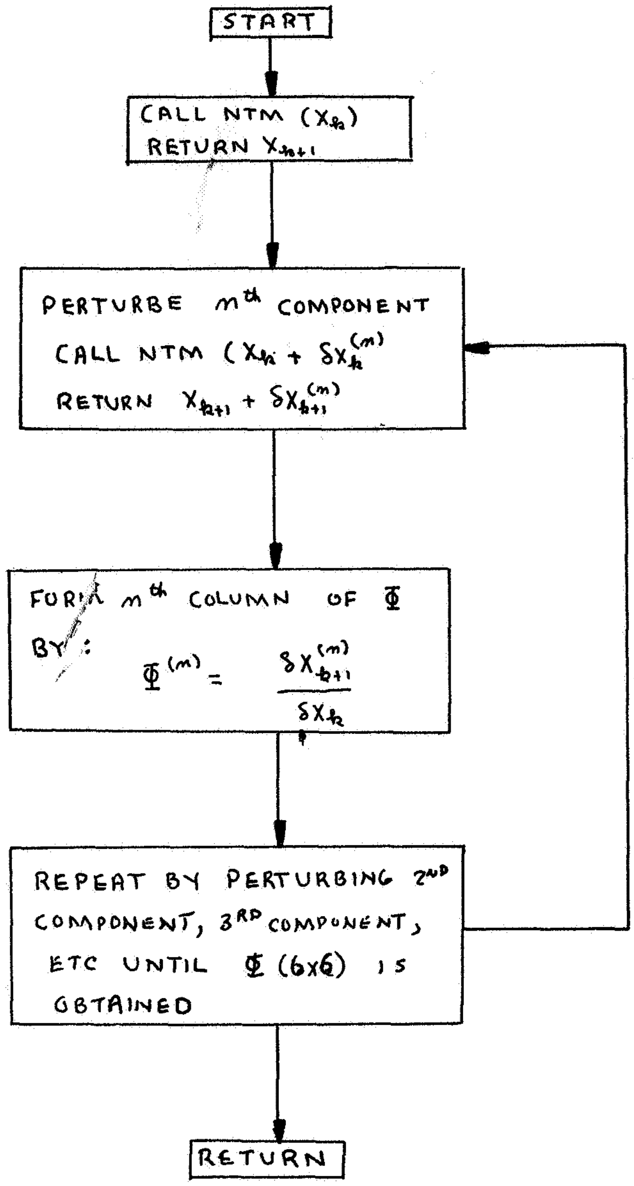
I/O	Fortran Name	Math Symbol	Definition
I	RI(6)	$\bar{X}_i$	Position and velocity of the vehicle at the beginning of the time interval.
I	RF(6)	$\bar{X}_f$	Position and velocity of the state vector at the end of the time interval.

Subprograms Required: NTM.

Approximate storage required (octal): 220.

Discussion: The numerical differencing technique used to compute the unaugmented portion of the state transition matrix consists of altering each component of the initial state vector in its turn and finding the final state vector corresponding to the new initial conditions. This results in obtaining six "new" state vectors at the final time, one corresponding to each altered initial component. The state transition matrix then consists of the differences in each component of the final state vector divided by the amount by which the initial component was altered.

Computational logic:



33. Subroutine NEWPGE

Purpose: This subroutine is used in conjunction with the virtual mass program when print-out of trajectory information is desired. NEWPGE prints the appropriate heading at the top of each page.

Calling sequence: CALL NEWPGE.

Input/output: None.

Subprograms required: None.

Approximate storage required (octal): 100.

Discussion: If a new page is desired in the printed output, NEWPGE is called. It allows the printer to skip to the top of the next page and prints the virtual mass heading. No computations relating to the technique are accomplished.

34. Subroutine NJEXN

Purpose: This program computes patched conic injection conditions corresponding to a mission specified by a launch date and planet, and a target data and planet.

Calling sequence: CALL NJEXN (JC3, JIJT, NDD, NTT, DDJD, TTJD, HHR1, HHV1, S).

I/O	Fortran name	Definition
I	JC3	Flag indicating whether biased (JC3=1) or unbiased (JC3=0) condition are generated
I	JINJT	Flag indicating whether injector time is updated (=0) or not (=1)
I	NDD	Index specifying launch planet
I	NTT	Index specifying target planet
I-O	DDJD	Julian date of launch (if J1NJT=0, output as injection time)
I	TTJD	Julian date of encounter
O	HHR1(3)	Injection position vector (launch planet ecliptic)
O	HHV1(3)	Injection velocity vector (launch planet ecliptic)
O	S(3)	Excess velocity at target planet

Subprograms required: AUX, CONST, EPHEM, HYPER, HYPSY, LAMB, ORB, OT2, PLANE, POSVL, TIME.

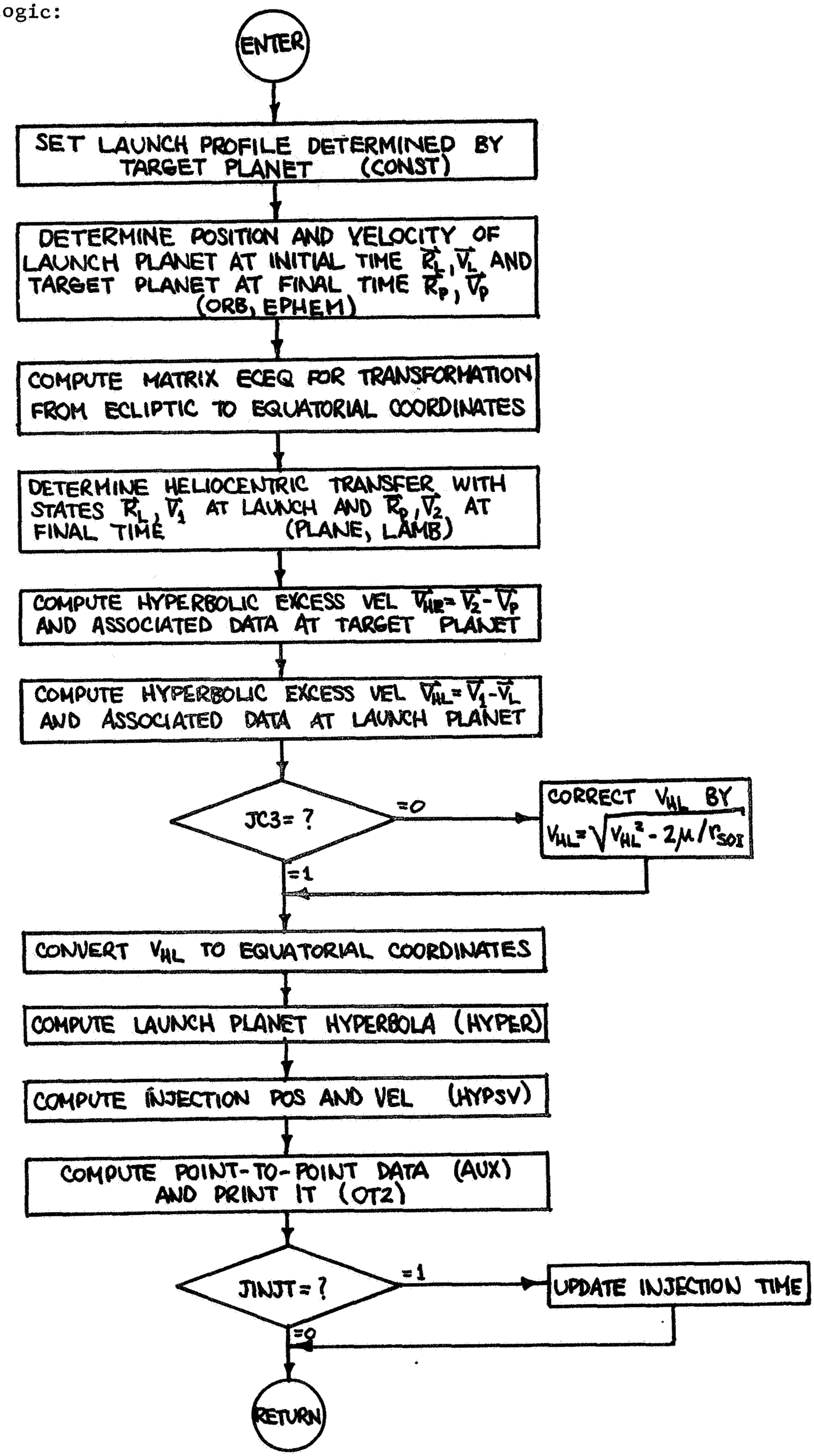
Approximate storage requied (octal): 1500.

Discussion: The NJEXN program is esstentially equivalent to the patched conic trajectory portion of the SPARC program developed in reference 1. A summary of the analytical basis of this program is provided in Volume II. Two main options have been added to the SPARC program in the NJEXN program.

NJEXN generates either of two quite similar sets of injection conditions. The first set specified by setting the flag JC3=1 is a good zero-iterate for n-body trajectory targeting (and is equivalent to the SPARC conditions); the second set determined when JC3=0 is closer to the condisions required for a targeted patched conic. The distinguishing computation of the two options is in the calculation of the velocity at infinity V_∞ before determining the near launch planet hyperbola (HYPER). If JC3=1, V_∞ is set equal to V_1 where V_1 is the magnitude of the difference between the heliocentric ellipse velocity vector and the launch planet orbital velocity vector. This is no error (in the patched conic sense) since the speed V_1 is actually desired at the sphere of influence of the planet where the heliocentric patching occurs. Thus, the actual desired velocity at infinity (for a patched conic) becomes $V_\infty = \sqrt{V_1^2 - 2\mu/r_{SI}}$. This is the excess velocity used when JC3=0, which yields improved injection velocities for a patched conic trajectory.

The second option is provided by a flag J1NJT. If J1NJT =0, the initial date is updated within the program to the injection time. If J1NJT=1 the initial time is left as its original value.

Computational logic:



35. Subroutine NTM

Purpose: NTM acts as an intermediate routine between the program calling for trajectory information and the virtual mass trajectory program itself. This subroutine sets various codes according to which trajectory is being run and what information is desired on return to the calling program.

Calling sequence: CALL NTM (RI, RF, NTMC, ICODE).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}_i$	State vector of the vehicle at the beginning of the time interval
O	RF(6)	$\overline{X}_f$	Position and velocity at the final time
I	NTMC	----	Nominal trajectory module code that determines which type of trajectory program is to be used. (Note: only the virtual mass technique is supplied with this program. However, with little effort any trajectory program may be added as an extra option)
I	ICODE	----	Internal code that determines which trajectory is being run and what information is desired.

Subprograms required: VMP.

Approximate storage requied (octal): 1630.

Discussion: NTM may be used to generate any of the three trajectories that are needed in the simulation mode of STEAP -- the original nominal trajectory, the most recent nominal trajectory, and the actual trajectory.

The input variable ICODE is used to distinguish between these trajectories. It is unimportant to the virtual mass technique which trajectory is being computed. However, it is important to keep them separated so that the proper codes are set that check

for approaching the sphere of influence of the target planet and reaching closest approach. It is also important to keep separate the conditions at which these occur for each trajectory. The following list descirbes ICODE completely.

ICODE = 3, NTM will check to see if the sphere of influence and/or closest approach has been reached on the actual trajectory. If not, VMP will check for these conditions and on encountering either, NTM places the conditions in special storage locations so they will be saved for future reference.

ICODE = 2, NTM performs the same operations as described above for the most recent nominal trajectory.

ICODE = 1, NTM again checks for sphere of influence and closest approach as above for the original nominal trajectory.

ICODE = 0, the only important information in this situation is the state vector at the end of the time interval. Therefore, NTM does not check to see if closest approach or sphere of influence is encountered. This might occur in numerical differencing, for example.

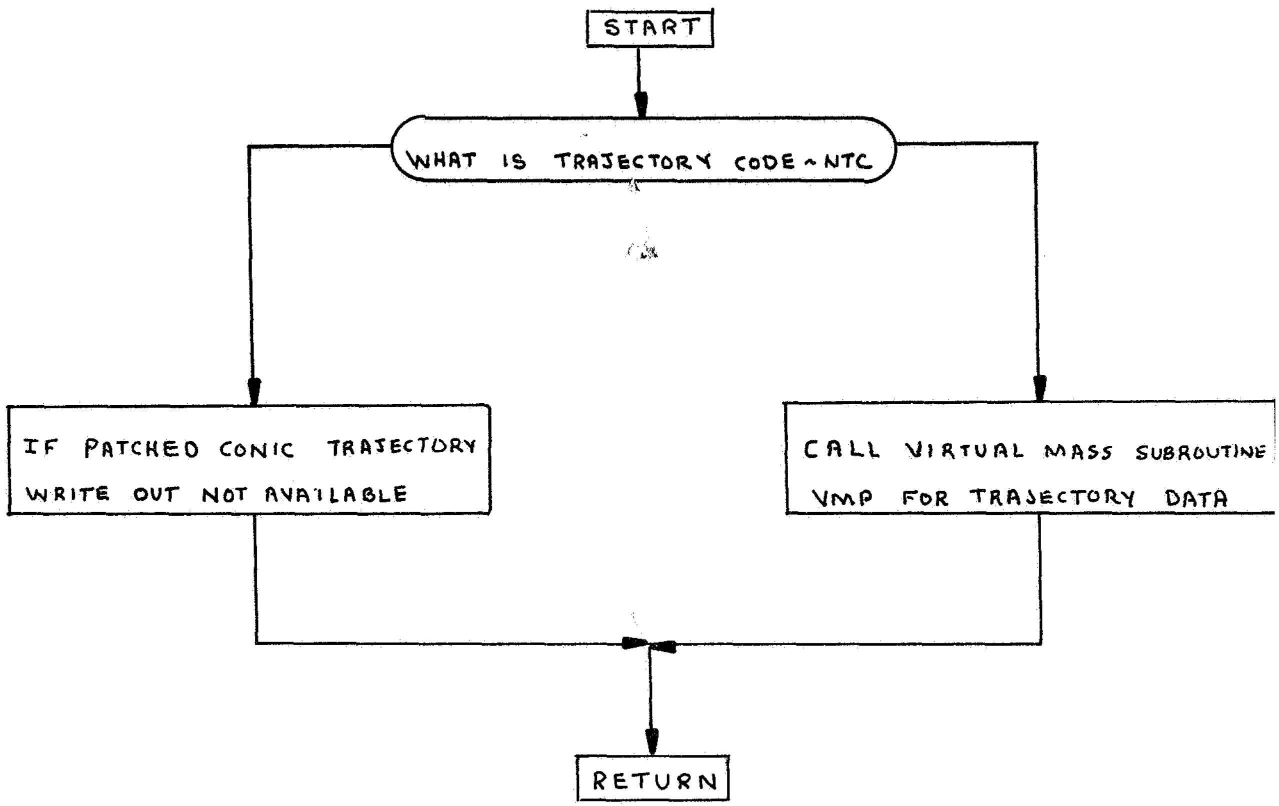
ICODE = -1, it is important to know if sphere of influence or closest approach is reached on the original nominal trajectory. However, it is not desired that the information be stored for future use. This situation occurs in the guidance event.

ICODE = -2, the same comments may be made as if ICODE = -1, except this is on the most recent nominal trajectory.

ICODE = -3, again, this value of ICODE is treated the same as is ICODE = -1, for the actual trajectory.

It should be pointed out that the only difference between the original nominal trajectory and the most recent nominal trajectory is that the most recent nominal trajectory may be updated at any time. However, the physical constants, the ephemeris, and other pertinent information are exactly the same in the two trajectories. This is not true of the actual trajectory. There may be biases in the ephemeris, in the gravitational constants of various bodies involved, or more bodies may be added to the analysis for the actual trajectory. NTM handles the logic involved for all of these options.

Computational logic:



36. Subroutine ORB

Purpose: The orbital elements -- inclination, longitude of ascending node, longitude of perihelion, eccentricity, and length of semimajor axis -- are computed for a specified planet at a given time.

Calling sequence: CALL ORB (IP, D).

Input/output:

I/O	Fortran name	Definition
I	IP	Code number of planet = 1, Sun = 2, Mercury = 3, Venus = 4, Earth = 5, Mars = 6, Jupiter = 7, Saturn = 8, Uranus = 9, Neptune = 10, Pluto = 11, Earth's Moon
I	D	Julian date, epoch 1900 of the time at which the elements are to be calculated.

Subprograms required: None.

Approximate storage required (octal): 250.

Discussion: The above mentioned elements are computed as a time series expansion as described in reference 1.

37. Subroutine OT2

Purpose: This program is responsible for converting the point-to-point injection conditions to convenient units and printing the resulting data in a format similar to that of the SPARC program.

Calling sequence: CALL OT2 (XL, XP, DDRM1, CCVM1, CCPSI, CCA, CCI, DDVM1, TTRM7, TTVM7, CCTA1, CCTA7, TL, TINJ, NTDO, TF, NTTT, C3, HHUM2, DLAQ, RALQ, RJ, HHVQM, PTH, VHP, DPA, RAP, HHE, DDAZ, TB, PHI, THI, RAI, AZI, TC, CCE).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	XL(6)		State of launch planet at initial date
I	XP(6)		State of target planet at target date
I	DDRM1	R_1	Heliocentric distance to launch planet at initial time
I	CCVM1	V_{LP}	Heliocentric speed of launch planet at initial time
I	CCPSI	ψ	Central angle of heliocentric conic
I	CCA	a	Semimajor axis of heliocentric conic
I	CCI	i	Inclination of heliocentric conic
I	DDVM1	V_1	Speed on heliocentric conic at initial time
I	TTRM7	R_2	Heliocentric distance to target planet at final time
I	TTVM7		Heliocentric speed of target planet at final time
I	CCTA1	ν_1	True anomaly on heliocentric conic at initial time
I	CCTA7	ν_2	True anomaly on heliocentric conic at final time
I	TL		Time of launch on launch date, hours

I	TINJ		Time of injection or launch date, hours
I	NTDD(5)		Year-month-day-hour-min of initial time
I	TF	t_f	Flight time
I	NTTT(5)		Year-month-day-hour-min of final time
I	C3	C_3	Energy of hyperbolic orbit
I	HHVM2	V_{HE}	Hyperbolic excess velocity at launch planet
I	DLAQ		Declination of HHVM2
I	RALQ		Right ascension of HHVM2
I	RJ		Injection radius
I	HHVQM		Injection velocity
I	PTH	Γ	Injection path angle
I	VHP	V_{HP}	Hyperbolic excess velocity at target planet
I	DPA	δ	Declination of VHP
I	RAP	Ω	Right ascension of VHP
I	HHE	e	Eccentricity of hyperbola
I	DDAZ	Σ	Launch azimuth
I	TB		Time from launch to injection
I	PHI	Φ_I	Injection latitude
I	THI	θ_I	Injection longitude
I	RAI	Θ_I	Injection right ascension
I	AZI	Σ_I	Injection azimuth
I	TC	t_c	Coast time
I	CCE	e	Heliocentric eccentricity

Subprograms required: None.

Approximate storage required (octal): 1000.

Discussion: The program simply prints the point to point data in a concurrent form.

38. Subroutine OUT1

Purpose: This program is responsible for the output of preliminary data before the numerical differencing cycle is begun.

Calling sequence: CALL OUT1 (ITARG, INJEK, N1TS, NB, IDAT1, S1, IDAT2, S2, IDAT3, S3, BDT, BDR, D1NCL, RCA, TOL1, TOL2, TOL3, ACC, RS, INPR, DELTP, NBOD, ISKEJ, AC, M1DI).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	ITARG		Targeting option flag
I	INJEK		Injection option flag
I	N1TS		Maximum allowable iterations at final level
I	NB(NBOD)		Indices of gravitational bodies
I	IDAT1(5), S1		Date of injection (year, month, day, hour, minute, sec)
I	IDAT2(5), S2		Date at sphere of influence
I	IDAT3(5), S3		Date of closest approach
I	BDT	$B \cdot T$	Impact plane parameter
I	BDR	$B \cdot R$	Impact plane parameter
I	D1NCL	i_{CA}	Inclination at closest approach
I	RCA	r_{CA}	Radius at closest approach
I	TOL1, TOL2, TOL3		Acceptable tolerances on target constraints
I	ACC		Final accuracy level
I	RS(6)		Injection state (position and velocity vectors)
I	1NPR		Integration increments between printouts in final integration

I	DELTP		Days between printouts in final integration
I	NBOD		Number of gravitational bodies
I	ISKEJ		Number of accuracy levels in targeting schedule
I	AC(ISKEJ)		Accuracy levels
I	MIDI		Number of iterations made at intermediate accuracy levels

Subprograms required: None.

Approximate storage required (octal): 1400.

Discussion: The program simply prints out specific program parameters.

39. Subroutine PARTL

Purpose: The partial of B·T and B·R with respect to the position and velocity of the vehicle are computed.

Calling sequence: CALL PARTL (R, V, B, BDT, BDR, PBT, PBR).

Input/output:

I/O	Fortran name	Definition
I	R(3)	Position of vehicle relative to planet
I	V(3)	Velocity of vehicle relative to planet
0	B	
0	BDT	B·T
0	BDR	B·R
0	PBT(6)	Partial of B·T with respect to R and V
0	PBR(6)	Partial of B·R with respect to R and V

Subprograms required: None.

Approximate storage required (octal): 330.

Discussion: This subroutine determines the partial derivatives of B·T and B·R with respect to the state vector. The general B-plane equations are given by,

$$B \cdot T = \frac{x\dot{y} - y\dot{x}}{\left(\dot{x}^2 + \dot{y}^2\right)^{1/2}}$$

$$B \cdot R = \frac{(x\dot{x} + y\dot{y})\dot{z} - \left(\dot{x}^2 + \dot{y}^2\right)z}{\left(\dot{x}^2 + \dot{y}^2\right)^{1/2}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right)^{1/2}}$$

Computational logic:

START

↓

$$B \cdot T = \frac{X\dot{Y} - Y\dot{X}}{(\dot{X}^2 + \dot{Y}^2)^{1/2}}$$

$$B \cdot R = \frac{(X\dot{X} + \dot{Y}Y)\dot{Z} - (\dot{X}^2 + \dot{Y}^2)Z}{(\dot{X}^2 + \dot{Y}^2)^{1/2}(\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2)^{1/2}}$$

$$B = (B \cdot T^2 + B \cdot R^2)^{1/2}$$

↓

$U2 = \dot{X}^2 + \dot{Y}^2$ $\qquad V2 = \dot{X}^2 + \dot{Y}^2 + \dot{Z}^2$

$U = (U2)^{1/2}$ $\qquad S = (V2)^{1/2}$

$UV = U * S$ $\qquad RU = X\dot{X} + Y\dot{Y}$

$U2PV2 = U2 + V2$ $\qquad UV3 = (UV)^3$

THE PARTIALS OF B·T AND B·R WITH RESPECT TO THE STATE VECTOR ARE:

$PBT_X = \dot{Y}/U$ $\qquad PBT_{\dot{X}} = -\dot{Y} * RU/U^3$

$PBT_Y = -\dot{X}/U$ $\qquad PBT_{\dot{Y}} = \dot{X} * RU/U^3$

$PBT_Z = 0.0$ $\qquad PBT_{\dot{Z}} = 0.0$

$PBR_X = \dot{X}\dot{Z}/UV$ $\qquad PBR_{\dot{X}} = \dot{Z}[U2(V2 * X - \dot{X} Z \dot{Z}) - \dot{X} * U2PV2 * RU]/UV3$

$PBR_Y = \dot{Y}\dot{Z}/UV$

$PBR_Z = -U/S$ $\qquad PBR_{\dot{Y}} = \dot{Z}[U2(V2 * Y - \dot{Y} Z \dot{Z}) - \dot{Y} * U2PV2 * RU]/UV3$

$PBR_{\dot{Z}} = U[X\dot{X} + Y\dot{Y} + Z\dot{Z}]/S^3$

↓

RETURN

40. Subroutine PCTM

Purpose: This routine computes the unaugmented portion of the state transition matrix using the patched-conic technique.

Calling sequence: CALL PCTM (RI).

Input/output:

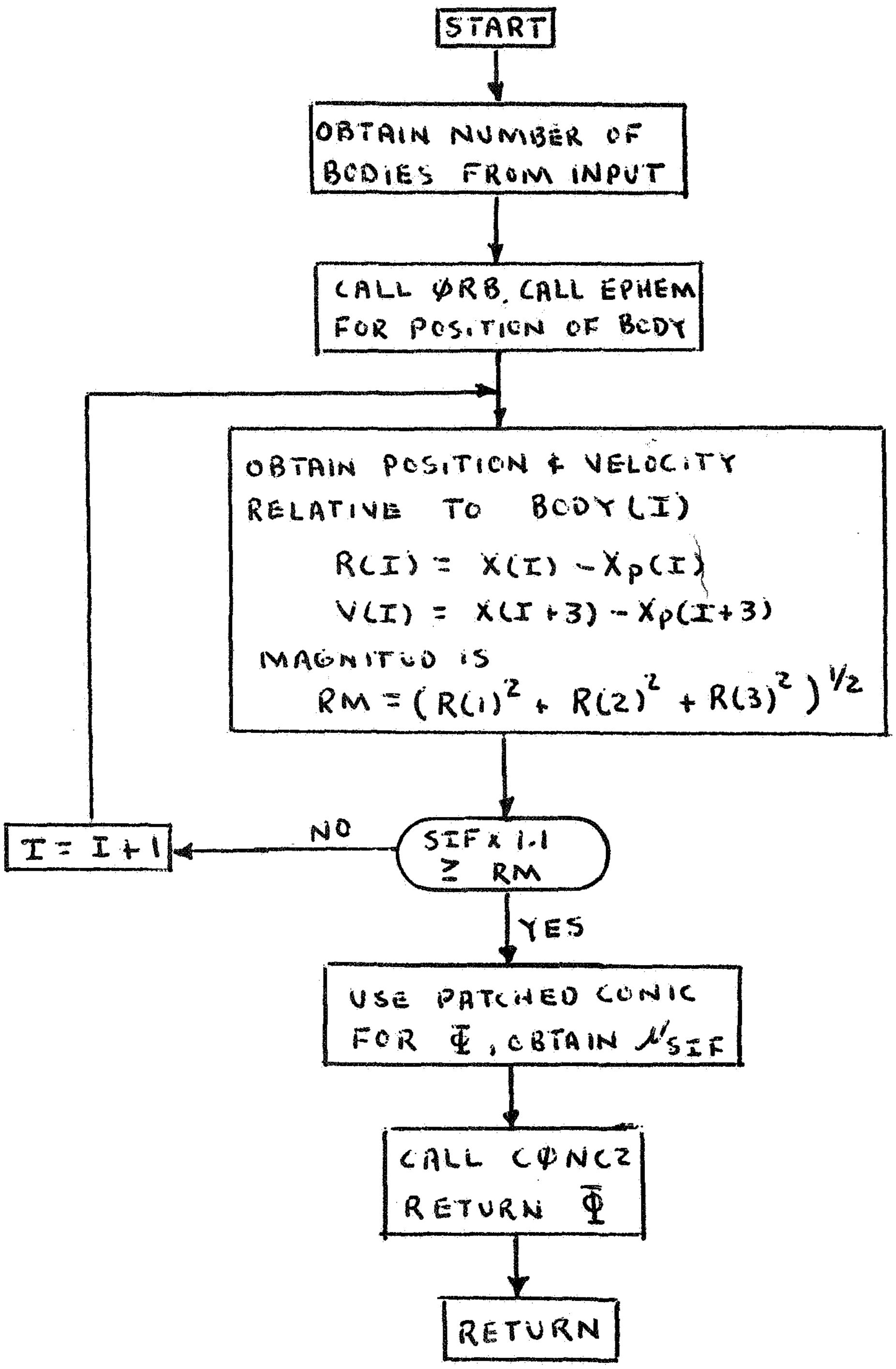
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Position and velocity of vehicle at beginning of time increment

Subprograms required: CONC2, EPHEM, ORB.

Approximate storage required (octal): 230.

Discussion: The subroutine checks each planet being considered in the analysis in turn and decides if the vehicle at the initial time is inside its sphere of influence. If it is not inside the sphere of influence of any planet the governing body is considered to be the Sun. After determining the governing body, PCTM calls the routine CONC2 to compute the unaugmented portion of the state transition matrix.

Computational logic:



41. Subroutine PECEQ

Purpose: This program computes the matrix defining the transformation from planet centered ecliptic coordinates to planet centered equatorial coordinates as a function of the particular planet and time.

Calling sequence: CALL PECEQ(NP,D,ECEQ).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	NP		Index of planet
I	D		Julian date (referenced to 1900)
0	ECEQ(3,3)	M_{ECEQ}	Transformation matrix

Subprograms required: EULMX.

Approximate storage required (octal): 400.

Discussion: The program sets four angles for the computation of the matrix. The angles and their definitions are:

XI - the inclination of the orbital plane to the ecliptic;

XL - the longitude of the ascending node of the orbital plane to the ecliptic;

XIQ - the inclination of the planet equator to the orbital plane;

XLQ - the longitude of the ascending node of the planet equator to the orbital plane.

The angles XI and XL are specified as functions of time for all planets. The angles XIQ and XLQ are set equal to zero for all planets except the Earth and Mars where they are set to nonzero constant values. These angles may be easily changed when better values are learned.

42. Subroutine PLANE

Purpose: This program calculates information pertaining to the heliocentric plane used in generating the injection conditions.

Calling sequence: CALL PLANE(XL, XP, HCA, HCW, HCN, NTYS).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	XL(6)		Initial state (position, velocity of launch planet
I	XP(6)		Final state of target planet
0	HCA	ψ	Heliocentric central angle
0	HCI		Inclination of heliocentric plane
0	HCW		Longitude of heliocentric plane
0	HCN(3)		Normal to heliocentric plane
0	NTYS		Flag set = 1 for $0 \leq \psi \leq 180$, = 2 for $180 \leq \psi \leq 360°$

Subprograms required: None.

Approximate storage required (octal): 400.

Discussion: This is an elementary program that is easily understood from the program listing.

43. Subroutine PLND

Purpose: The portion of the state transition matrix corresponding to the state vector in which the ephemeris biases of the target planet are augmented is computed in PLND.

Calling sequence: CALL PLND (RI, RF).

Input/output:

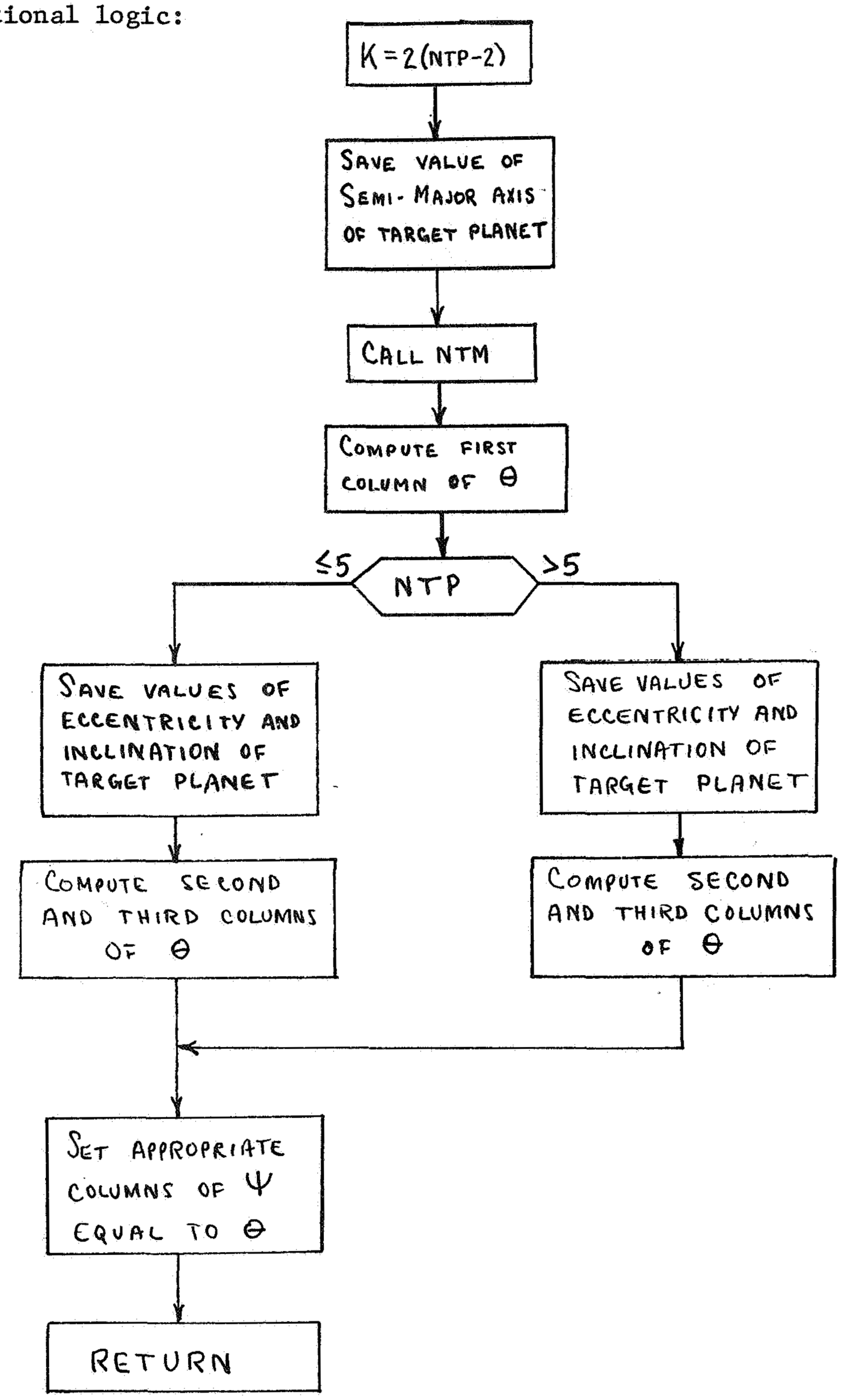
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}_i$	Position and velocity of vehicle at beginning of interval.
I	RF(6)	$\overline{X}_f$	Position and velocity of vehicle at end of interval.

Subprograms required: NTM.

Approximate storage required (OCTAL): 440.

Discussion: The numerical differencing method is used to generate the 6 x 3 portion of the state transition matrix corresponding to the augmented state vector. The ephemeris biases that are augmented are the semimajor axis, eccentricity, and inclination. The values of these used by the virtual mass program are altered in turn by previously specified increments and the differences in the final state vector are noted. Finally, the state transition matrix is computed as the differences in each component of the state vector, divided by the amount by which the appropriate constant was altered.

Computational logic:



44. Subroutine POSVL

Purpose: This program calculates the vector position and velocity corresponding to a specified mean anomaly on a specified ellipse.

Calling sequence: CALL POSVL(A, E, XI, WC, W, AM, WP, RP, R, VP, V, GMS).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	A	a	Semimajor axis
I	E	e	Eccentricity
I	XI	i	Inclination
I	WC	Ω	Longitude of ascending node
I	W	ω	Argument of periapsis
I	AM	M	Mean anomaly
I	WP(3)	$\hat{W}$	Normal to plane
O	RP(3)		Position vector (heliocentric ecliptic)
O	R		Position magnitude
O	VP(3)		Velocity vector (heliocentric ecliptic)
O	V		Speed
I	GMS	μ_S	Gravitational constant of sun

Subprograms required: None.

Approximate storage required (octal): 600.

Discussion: The program solves Kepler's equation iteratively to compute the eccentric anomaly. The vector position and velocity are then computed by standard conic formulas.

45. Subroutine PRED

Purpose: This subroutine is responsible for the logic at a prediction event in the error analysis mode of STEAP. It computes the matrix of uncertainties in the state vector when a prediction is made from a stated time to another specified time.

Calling sequence: CALL PRED (RI, TEVN).

Input/output:

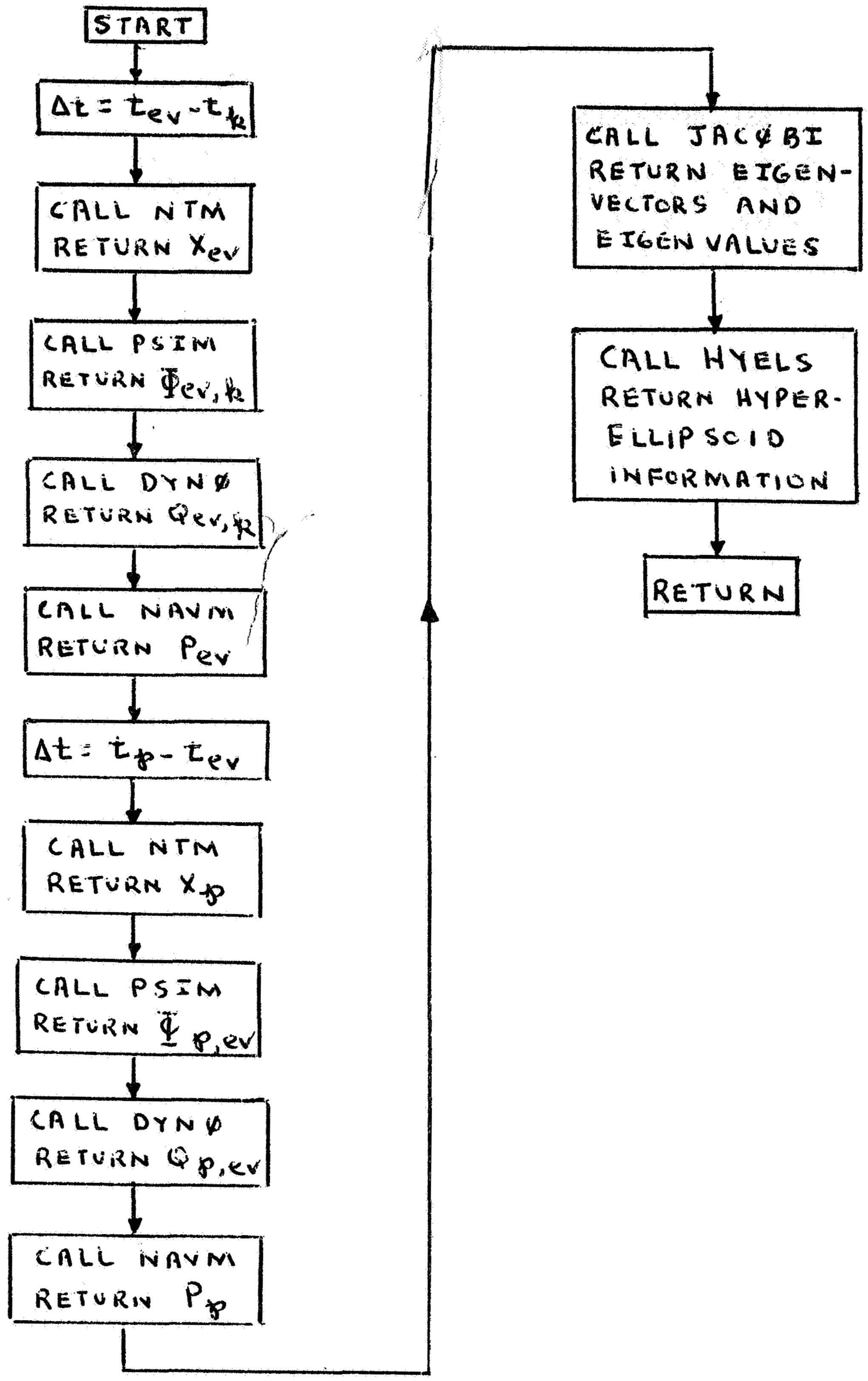
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}$	Position and velocity of the vehicle at the time of the last measurement or event.
I	TEVN	t_{ev}	The trajectory time of the prediction event.

Subprograms required: DYNO, HYELS, JACOBI, NAVM, NTM, PSIM.

Approximate storage required (octal): 4260.

Discussion: In PRED, the covariance matrix is propagated forward to the time of the prediction event, t_{ev}, as in EIGEN. At this time, however, an additional computation is made that propagates it forward to the time to which one is predicting, t_{PT}. This covariance matrix is then diagonalized and the eigenvalues and eigenvectors are printed. The program then returns to the basic cycle for processing of the next measurement or event.

Computational logic:



46. Subroutine PRESIM

Purpose: The routine contains the logic for a prediction event in the simulation mode of STEAP.

Calling sequence: CALL PRESIM (RI, TEVN, RI1).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Original nominal state vector of the vehicle at the time of the last measurement or event.
I	TEVN	t_{ev}	Time of the prediction event.
I	RI1(6)	$\tilde{X}$	The most recent nominal state vector at the time of the last measurement or event.

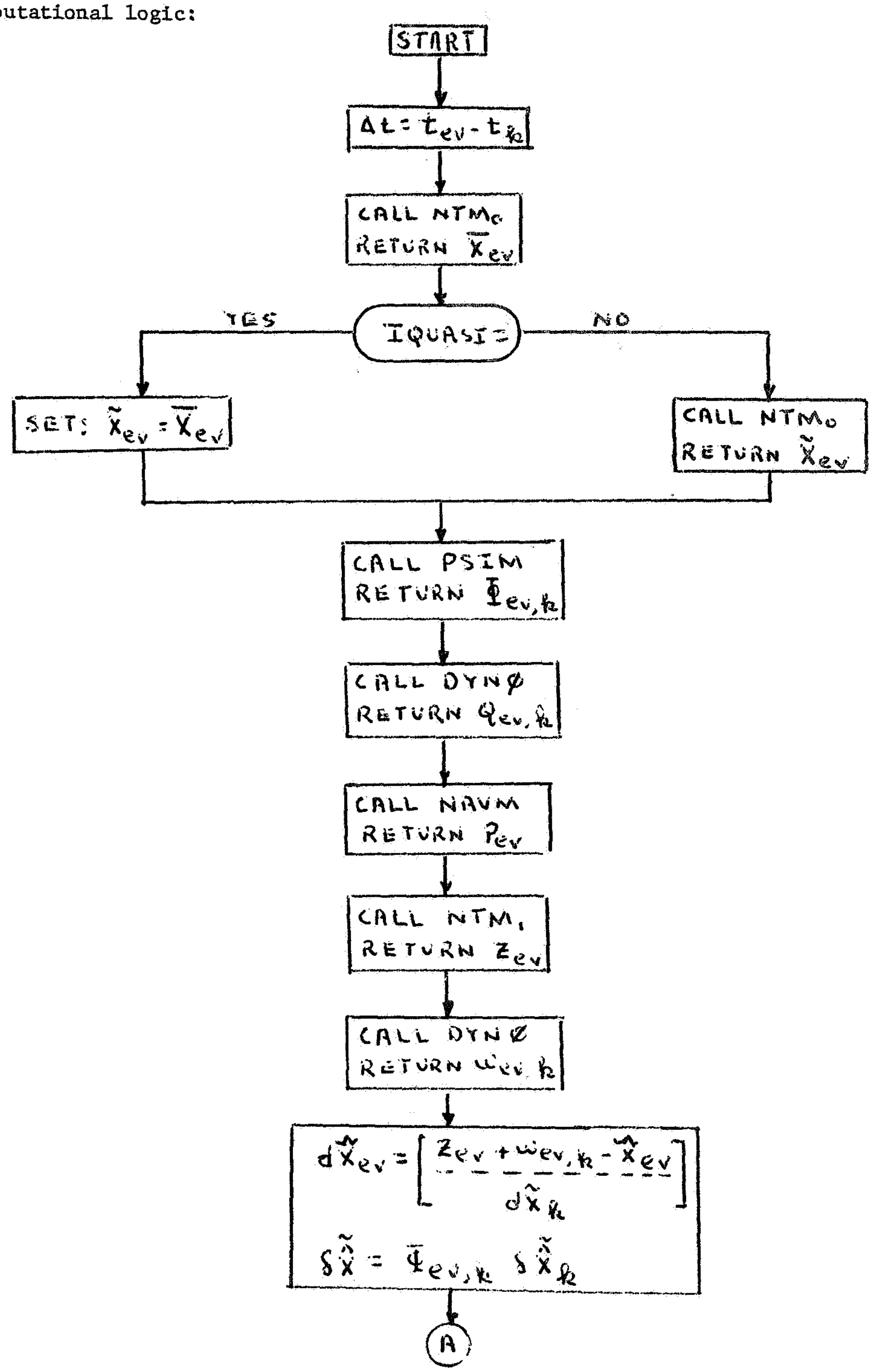
Subprograms required: DYNO, HYELS, JACOBI, NAVM, NTM, PSIM.

Approximate storage required (octal): 4550.

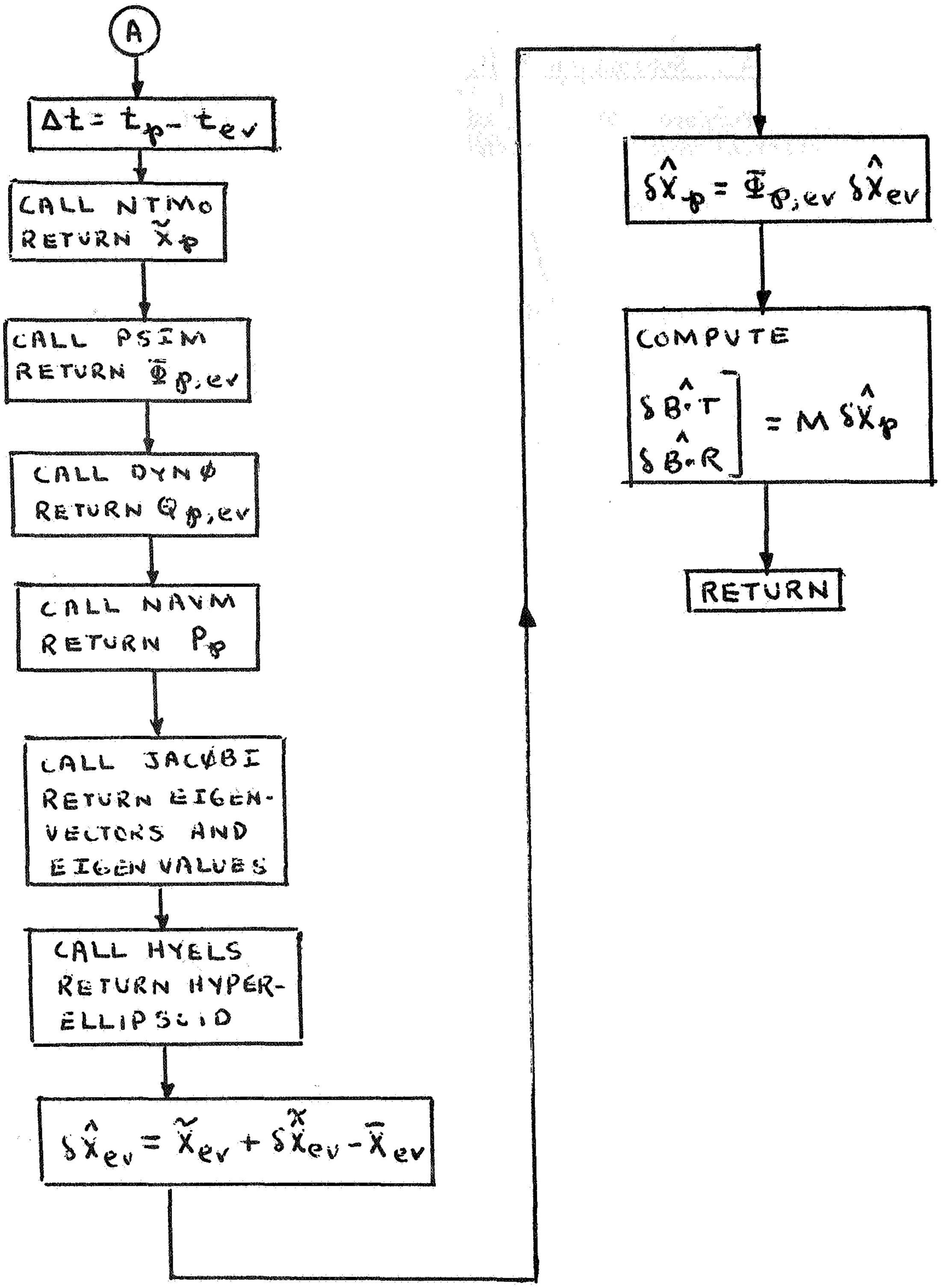
Discussion: The covariance matrix is propagated forward from the time of the last measurement or event to the time of the prediction event. This matrix is diagonalized and the eigenvalues, eigenvectors, and hyperellipsoids are computed and printed. The correlation coefficient matrix is also printed. Then, the covariance matrix is propagated forward to the prediction time, t_{PT}, and again diagonalized. The routine then returns control to the basic cycle.

The covariance matrices mentioned above are based on the most recent nominal trajectory for the simulation mode rather than the original trajectory used in PRED for the error analysis mode.

Computational logic:



Computational logic (concluded):



47. Subroutine PRINT

Purpose: This subroutine is responsible for printing the virtual mass information as desired.

Calling sequence: CALL PRINT

Input/output: None.

Subprograms required: NEWPGE, SPACE, TIME.

Approximate storage required (octal): 1210.

Discussion: The printed output has four sections:

1) Spacecraft information giving the spacecraft inertial trajectory;

2) Ephemeris data that prints the position and velocity of each planet;

3) Spacecraft relative trajectories in which the position and velocity of the spacecraft relative to each planet is printed;

4) The virtual mass information is printed in which the position and velocity of the virtual mass is given in addition to the position and velocity of the spacecraft relative to the virtual mass, the Kepler vector, the eccentricity vector, the virtual mass magnitude, and the magnitude rate;

5) Finally, the position and velocity of the virtual mass relative to each of the planets is printed.

48. Subroutine PRINT1

Purpose: This routine prints a summary of the trajectory generated by the trajectory mode of STEAP.

Calling sequence: CALL PRINT1 (RF).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RF(6)	$\overline{X}$	Position and velocity of the vehicle at the final time.

Subprograms required: TIME.

Approximate storage required (octal): 1360.

Discussion: The initial and final position and velocity of the vehicle is printed together with the position and velocity of the vehicle relative to each planet at the final time. The conditions at closest approach and on encountering the sphere of influence are printed.

49. Subroutine PRINT3

Purpose: The pertinent information at the end of each measurement is printed in this routine.

Calling sequence: CALL PRINT3 (MMCODE, NR).

Input/output:

I/O	Fortran name	Definition
I	MMCODE	Code that determines which type of measurement was made.
I	NR	Number of rows in the observation matrix H.

Subprograms required: EPHEM, ORB, TIME.

Approximate storage required (octal): 2320.

Discussion: For a listing of the printed output received from this routine refer to Chapter III.

50. Subroutine PRINT4

Purpose: This subroutine prints all necessary data at the end of each measurement in the simulation mode.

Calling sequence: CALL PRINT4 (MMCODE, NR).

Input/output:

I/O	Fortran name	Definition
I	MMCODE	Measurement code that determines which type of measurement was taken
I	NR	Number of rows in the observation matrix

Subprograms required: EPHEM, ORB, TIME.

Approximate storage required (octal): 3710.

Discussion: An outline of the printed output for which this routine is responsible is given in Chapter III.

51. Subroutine PRNTS3

Purpose: This subroutine prints a summary of the error analysis mode.

Calling sequence: CALL PRNTS3 (RF).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RF(6)	$\bar{X}$	Position and velocity of vehicle at final time

Subprograms required: TIME.

Approximate storage required (octal): 2760.

Discussion: See Chapter III for a detailed account of the printed output generated by PRNTS3.

52. Subroutine PRNTS4

Purpose: A printout summary of the simulation mode is presented by this subroutine.

Calling sequence: CALL PRNTS4 (RF, RF1).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RF(6)	$\overline{X}$	Position and velocity of the vehicle on the original nominal trajectory at the final time.
I	RF1(6)	$\tilde{X}$	Position and velocity of the vehicle on the most recent nominal trajectory at the final time.

Subprograms required: EPHEM, ORB, TIME.

Approximate storage required (octal): 6210.

Discussion: For a description of the printed output see Chapter III.

53. Subroutine PSIM

Purpose: The logic for computation of state transition matrices is provided by this routine.

Calling Sequence: CALL PSIM (RI, RF, ISTMC).

Input/output:

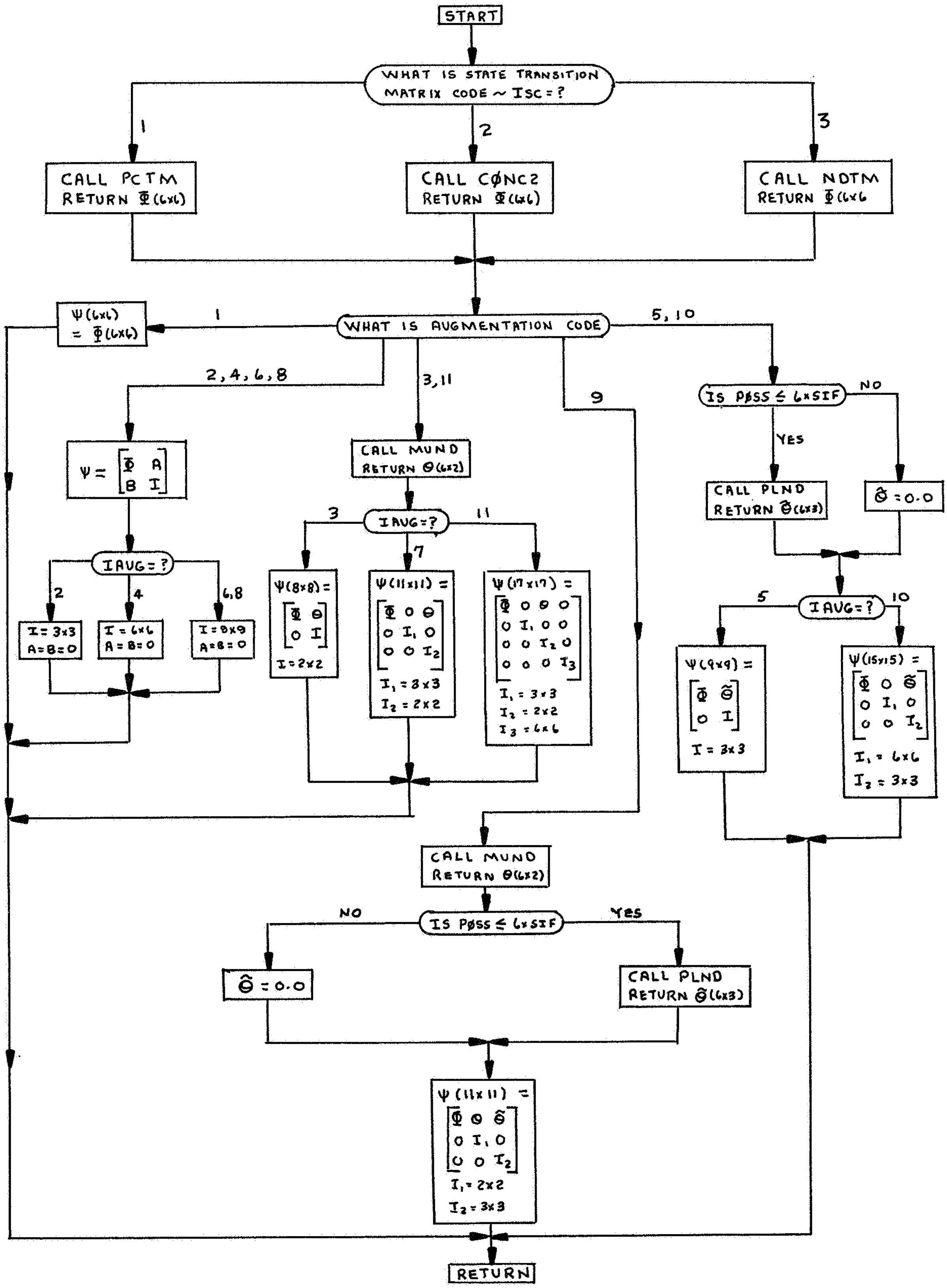
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\overline{X}_i$	Position and velocity of the vehicle at the beginning of the interval
I	RF(6)	$\overline{X}_f$	Position and velocity of the vehicle at the end of the interval.
I	ISTMC		A code specifying which technique is to be used to compute the unaugmented portion of the state transition matrix (see Chapter II for more details).

Subprograms Required: CONC2, EPHEM, MUND, NDTM, ORB, PCTM, PLND.

Approximate storage required (octal): 360.

Discussion: A decision is made through the use of ISTMC as to which technique will be used to compute the unaugmented portion of the state transition matrix. Then the proper subroutine is called to accomplish this. If the gravitational constants of the Sun and the target planet have been augmented to the state, the subroutine MUND is called to calculate the corresponding portion of the state transition matrix. Finally, a check is made to determine if the ephemeris biases of the target planet have been augmented to the state and, if so, it is determined whether the distance of the vehicle from the target planet is less than six times its sphere of influence, in which case the subroutine PLND is called to compute the final portion of the state transition matrix. If the vehicle is farther from the target planet than six times its sphere of influence, that portion of the state transition matrix is considered zero.

Computational logic:



54. Subroutine QUASI

Purpose: This subroutine contains the logic required for the quasi-linear filtering event in the simulation mode whose purpose it is to update the most recent nominal trajectory so that it might correspond more closely with the actual trajectory.

Calling Sequence: CALL QUASI (RI, TEVN, RI1).

Input/output:

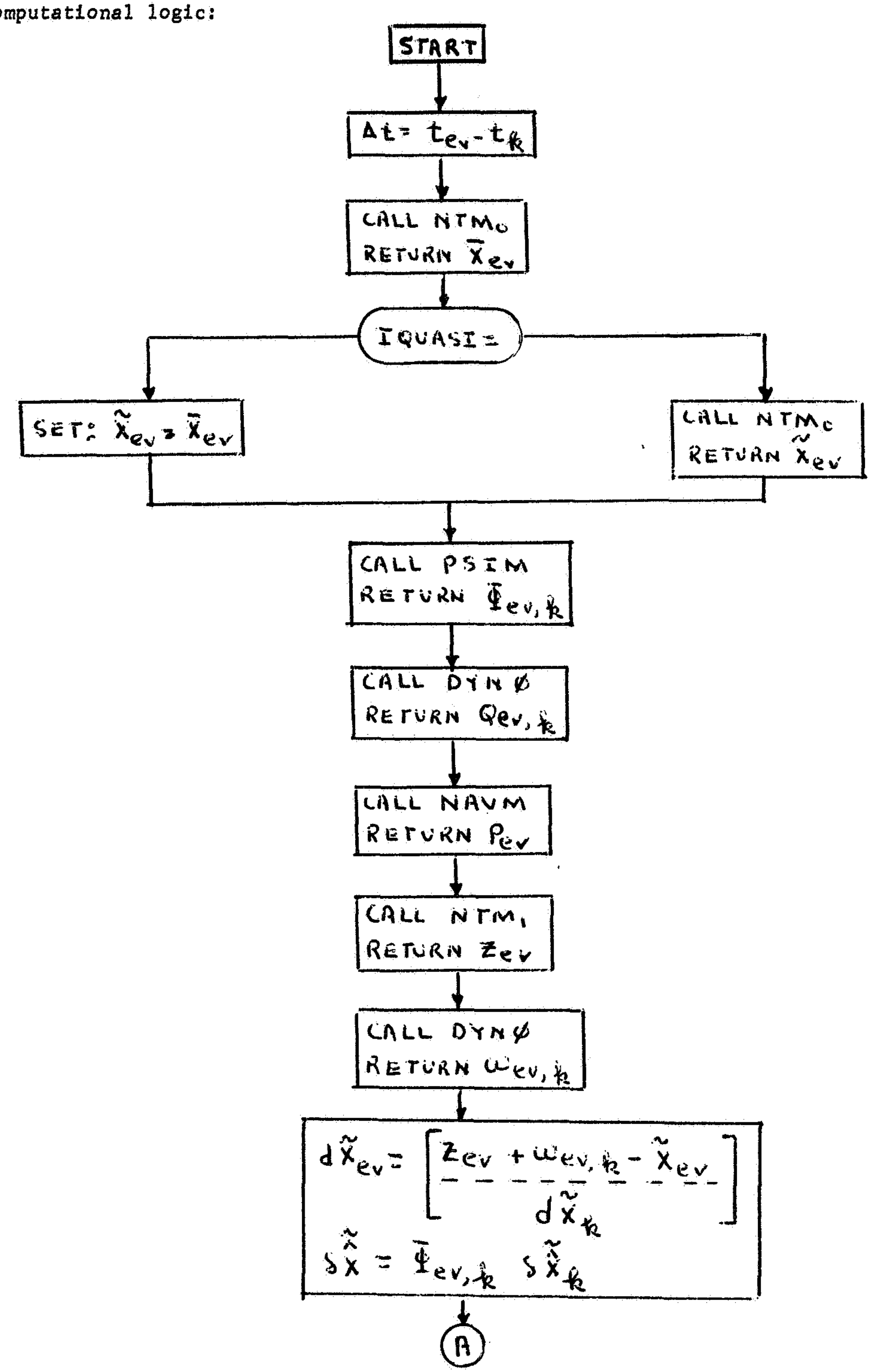
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Position and velocity of the vehicle on the original nominal trajectory at the time of the last measurement or event
I	TEVN	t_{ev}	Trajectory time of the quasi-linear filtering event.
I	RI1(6)	$\tilde{X}$	Position and velocity of the vehicle on the most recent nominal trajectory at the time of the last measurement or event.

Subprograms required: DYNO, HYELS, JACOBI, NAVM, NTM, PSIM.

Approximate storage required (octal): 2270.

Discussion: At a quasi-linear filtering event, the original nominal trajectory is updated by using the most recent nominal estimate. This event is provided as a method to help combat divergence due to the possible invalidity of the linearizing assumption that is the basis for the estimation algorithm. The general equations for updating the original nominal are provided in Volume II of this report.

Computational logic:



Computational logic (concluded):

(A)

COMPUTE;

$$\tilde{X}_{ev} = \tilde{X}_{ev} + \delta\hat{\tilde{X}}_{ev}$$

$$d\tilde{X}_{ev} = \begin{bmatrix} Z_{ev} + W_{ev,k} - \tilde{X}_{ev} \\ \hline d\tilde{X}_{ev} \end{bmatrix}$$

$$\delta\hat{\tilde{X}}_{ev} = \begin{bmatrix} 0 \\ \hline \delta\hat{\tilde{X}}_{ev} \end{bmatrix}$$

$$P_{ev} = P_{ev}$$

RETURN

NOTE: ONLY 6 COMPONENTS OF 6X1 $\tilde{X}_{ev}$ AND FIRST 6 COMPONENTS OF NDIM x 1 $d\check{X}_{ev}$ AND $\delta\hat{\tilde{X}}_{ev}$ CHANGE AT A QUASI-LINEAR FILTERING EVENT.

55. Subroutine RNUM

Purpose: RNUM is a function subprogram whose purpose it is to return random numbers on a normal distribution with mean zero and standard deviation.

Calling sequence: A = RNUM (SIGMA)

Input/output:

I/0	Fortran Name	Math symbol	Definition
I	SIGMA	σ	Standard deviation of the normal distribution
0	RNUM	X	Randomly distributed number from the population described above.

Subprograms required: None.

Approximate storage required (octal): 130.

Discussion: The method used here to generate a randomly distributed number from a normal distribution with mean zero and standard deviation, σ, is to compute twelve random numbers between 0 and 1 (many routines that are statistically consistent may be found that accomplish this function).

$$\text{Then RNUM} = \left[\sum_{i=1}^{12} X_i - 6\right]\sigma.$$

56. Subroutine SCHED

Purpose: The routine determines what type of measurement is to be taken next and at what time it will occur.

Calling sequence: CALL SCHED (T1, T2, MMCODE).

Input/output:

I/O	Fortran name	Definition
I	T1	Present trajectory time.
O	T2	Trajectory time at which the next measurement occurs.
O	MMCODE	Measurement model code (see Chapter II for details)

Subprograms required: None.

Approximate storage required (octal): 40.

Discussion: Chapter II describes the means by which the measurement schedule is input. A short part of the DATA subprogram then arranges these in the order in which they are to occur. SCHED simply finds the next measurement after T1.

57. Subroutine SPACE

Purpose: This subroutine is used in the virtual mass program for computing the trajectories. It keeps a count of the total number of lines that have been written on a given page and decides when to start a new page.

Calling sequence: CALL SPACE (LINES).

Input/output:

I/O	Fortran name	Definition
I	LINES	Number of line that will be written in the next output statement

Subprograms required: NEWPGE.

Approximate storage required (octal): 30.

Discussion: If the total number of lines will exceed the maximum lines per page SPACE calls NEWPGE.

58. Subroutine STAPARL

Purpose: The partial derivatives of station location errors are computed.

Calling sequence: CALL PARTL (AL, ALON, ALAT, PAT2, VEC, PA).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	AL	r	Altitude of the station
I	ALON	θ	Longitude of the station
I	ALAT	φ	Latitude of the station
I	PAT2	----	
I	VEC(6)	$\overline{X}$	Position of the vehicle relative to the station
O	PA(6,3)	----	Partial of altitude, latitude, and longitude with respect to VEC.

Subprograms required: None.

Approximate storage required (octal): 300.

Discussion: This subroutine computes the partial derivatives for station location errors when the geocentric radius, latitude, and longitude are included in the augmented state vector.

From the general equations for defining the position and velocity of a station on a rotating Earth, (from TRAKM subroutine) the following partial derivatives are obtained for any of the three stations:

$$\text{Let } G = \varphi + \omega\,(t - T)$$

$$-\frac{\partial \overline{X}}{\partial R} = -\cos\theta \cos G$$

$$-\frac{\partial \overline{X}}{\partial \theta} = R \sin\theta \cos G$$

$$- \frac{\partial \overline{X}}{\partial \varphi} = R \cos \theta \sin G$$

$$- \frac{\partial \overline{Y}}{\partial R} = -[\sin \epsilon \sin \theta + \cos \varepsilon \cos \theta \sin G]$$

$$- \frac{\partial \overline{Y}}{\partial \theta} = R \cos \epsilon \sin \theta \sin G - R \cos \epsilon \cos \theta$$

$$- \frac{\partial \overline{Y}}{\partial \varphi} = -R \cos \epsilon \cos \theta \cos G$$

$$- \frac{\partial \overline{Z}}{\partial R} = \sin \epsilon \cos \theta \sin G - \cos \epsilon \sin \theta$$

$$- \frac{\partial \overline{Z}}{\partial \theta} = -[R \sin \epsilon \sin \theta \sin G + R \cos \epsilon \cos \theta]$$

$$- \frac{\partial \overline{Z}}{\partial \varphi} = R \sin \epsilon \cos \theta \cos G$$

$$- \frac{\partial \dot{\overline{X}}}{\partial R} = \omega \cos \theta \sin G$$

$$- \frac{\partial \dot{\overline{X}}}{\partial \theta} = -\omega R \sin \theta \sin G$$

$$- \frac{\partial \dot{\overline{X}}}{\partial \varphi} = \omega R \cos \theta \cos G$$

$$- \frac{\partial \dot{\overline{Y}}}{\partial R} = -\omega \cos \theta \cos \varepsilon \cos G$$

$$- \frac{\partial \dot{\overline{Y}}}{\partial \theta} = \omega R \cos \epsilon \sin \theta \cos G$$

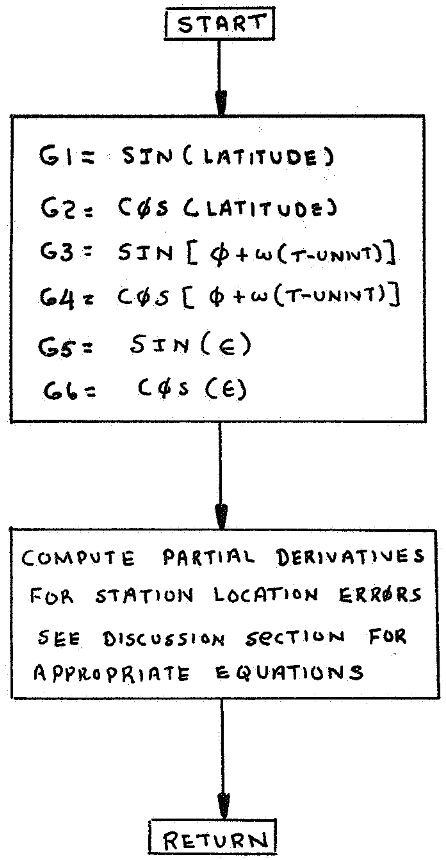
$$- \frac{\partial \dot{\overline{Y}}}{\partial \varphi} = \omega R \cos \epsilon \cos \varphi \sin G$$

$$- \frac{\partial \dot{\overline{Z}}}{\partial R} = \omega \sin \epsilon \cos \varphi \cos G$$

$$-\frac{\partial \dot{Z}}{\partial \theta} = -\omega R \sin \epsilon \sin \theta \cos G$$

$$-\frac{\partial \dot{Z}}{\partial \varphi} = -\omega R \sin \epsilon \cos \theta \sin G$$

Computational logic:



59. Subroutine TIME

Purpose: If the Julian date is supplied to this routine, the corresponding calendar date will be returned. Alternately, the Julian date, epoch 1900, will be returned if the calendar date is supplied.

Calling sequence: CALL TIME (DAY, IYR, MO, IDAY, IHR, MIN, SEC, ICODE).

Input/output:

I/O	Fortran name	Definition
I/O	DAY	Julian date, epoch Jan. 0, 1900.
O/I	IYR	Calendar year
O/I	MO	Calendar month
O/I	IDAY	Calendar day
O/I	IHR	Hour of day
O/I	MIN	Minutes
O/I	SEC	Seconds
I	ICODE	An internal code that determines which of the above options is exercised.

Subprograms required: None.

Approximate storage required (octal): 250.

Discussion: This subroutine will convert from Julian date, epoch January 0, 1900, to calendar date or from calendar date to Julian date depending on the value of ICODE as mentioned above.

ICODE = 0 indicates the calendar date is supplied, Julian date will be returned.

ICODE = 1 Julian date is given and calendar date will be returned.

Note: In the calendar date as supplied to this routine, the year, month, day, hour, and minutes are considered integers. However, the number of seconds is returned or supplied as a real (floating point) number. Thus, a calendar date might be July 10, 5 hr, 6 min, 3.22 sec, 1972.

For a discussion of the equations used in these conversions see reference 1.

60. Subroutine TRAKM

Purpose: The observations and the observation matrix for a given type of measurement is computed by this routine.

Calling sequence: CALL TRAKM (HECV, ITRK, NR, IOBS, VECTOR).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	HECV(6)	$\bar{X}$	Position and velocity of the vehicle at the time of the measurement.
I	ITRK		Code that determines what type of measurement is being made. (Note: this variable is called MMCODE elsewhere in STEAP).
0	NR	n	Number of rows in the observation matrix.
I	IOBS		Internal code that states whether only the observation is desired or if both the observation and the observation matrix are to be computed.
0	VECTOR(4)	$\tilde{Y}$	Observation that is made.

Subprograms required: EPHEM, ORB, STAPARL.

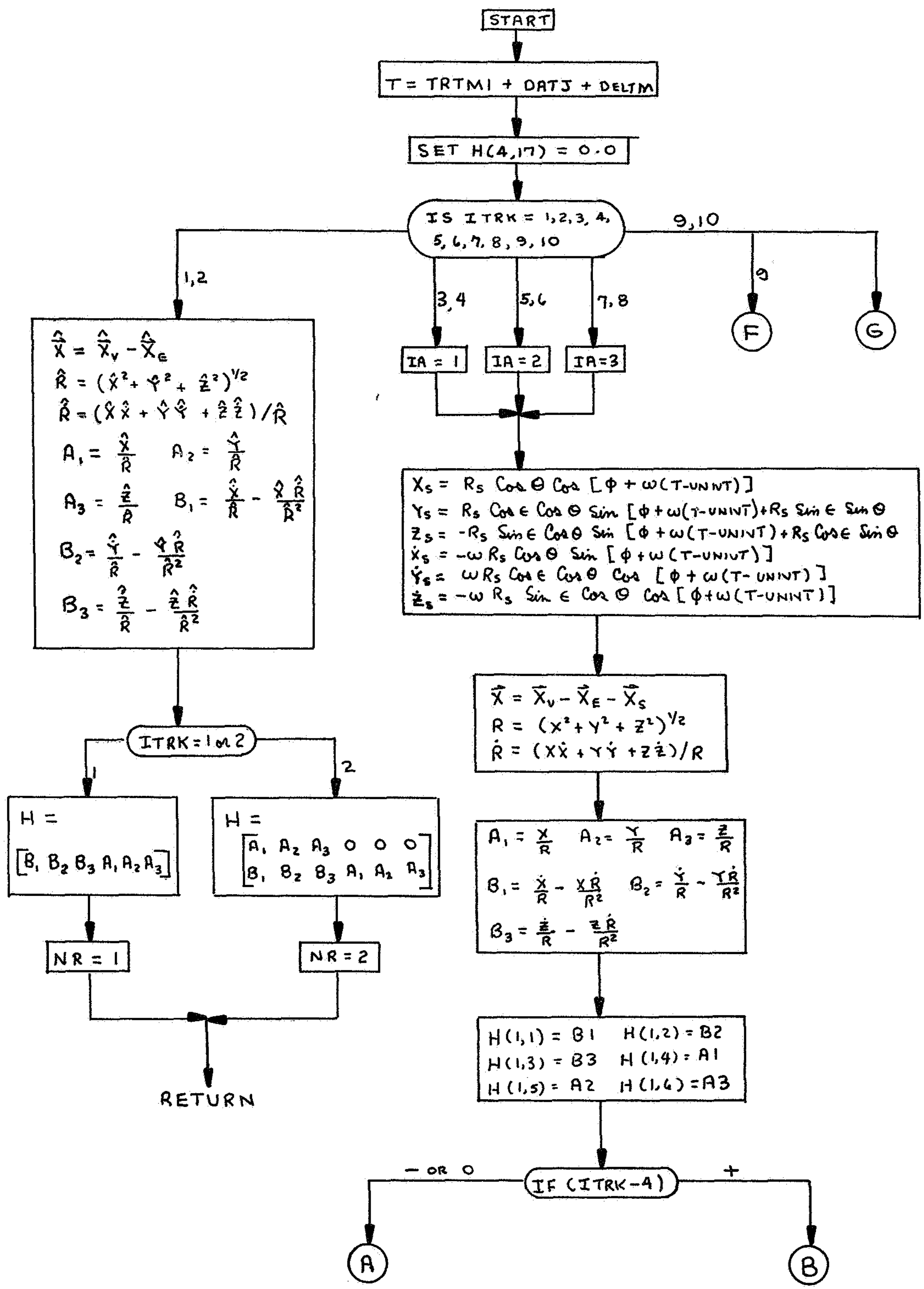
Approximate storage required (octal): 2310.

Discussion: The equations used to compute the observation matrix are discussed in Volume II of this document. The observation matrix is needed for both the error analysis and simulation modes when a measurement is taken. The observation itself is needed only in the simulation mode to compute the estimated and actual measurements. Which of these options is exercised is determined by IOBS. When IOBS = 0, the observation matrix is needed. Therefore, the observation itself is not placed in VECTOR. However, some "dummy" vector should appear in the calling sequence regardless of whether it is used.

When IOBS = 1, the estimated observation will be returned in VECTOR to the simulation mode of STEAP.

When IOBS = 2, the actual observation will be returned to STEAP. The value of VECTOR depends primarily on the state vector of the vehicle, HECV, at the time the observation is being made. The only reason there is a need to distinguish between the estimated and actual observations is that in considering the actual observation a bias in the station locations may be taken into consideration. See Chapter II for input options.

Computational logic:



$$\hat{\vec{X}} = \hat{\vec{X}}_V - \hat{\vec{X}}_E$$
$$\hat{R} = (\hat{X}^2 + \hat{Y}^2 + \hat{Z}^2)^{1/2}$$
$$\dot{\hat{R}} = (\hat{X}\dot{\hat{X}} + \hat{Y}\dot{\hat{Y}} + \hat{Z}\dot{\hat{Z}})/\hat{R}$$
$$A_1 = \frac{\hat{X}}{\hat{R}} \quad A_2 = \frac{\hat{Y}}{\hat{R}} \quad A_3 = \frac{\hat{Z}}{\hat{R}} \quad B_1 = \frac{\dot{\hat{X}}}{\hat{R}} - \frac{\hat{X}\dot{\hat{R}}}{\hat{R}^2}$$
$$B_2 = \frac{\dot{\hat{Y}}}{\hat{R}} - \frac{\hat{Y}\dot{\hat{R}}}{\hat{R}^2} \quad B_3 = \frac{\dot{\hat{Z}}}{\hat{R}} - \frac{\hat{Z}\dot{\hat{R}}}{\hat{R}^2}$$

$$H = \begin{bmatrix} B_1 & B_2 & B_3 & A_1 & A_2 & A_3 \end{bmatrix}$$

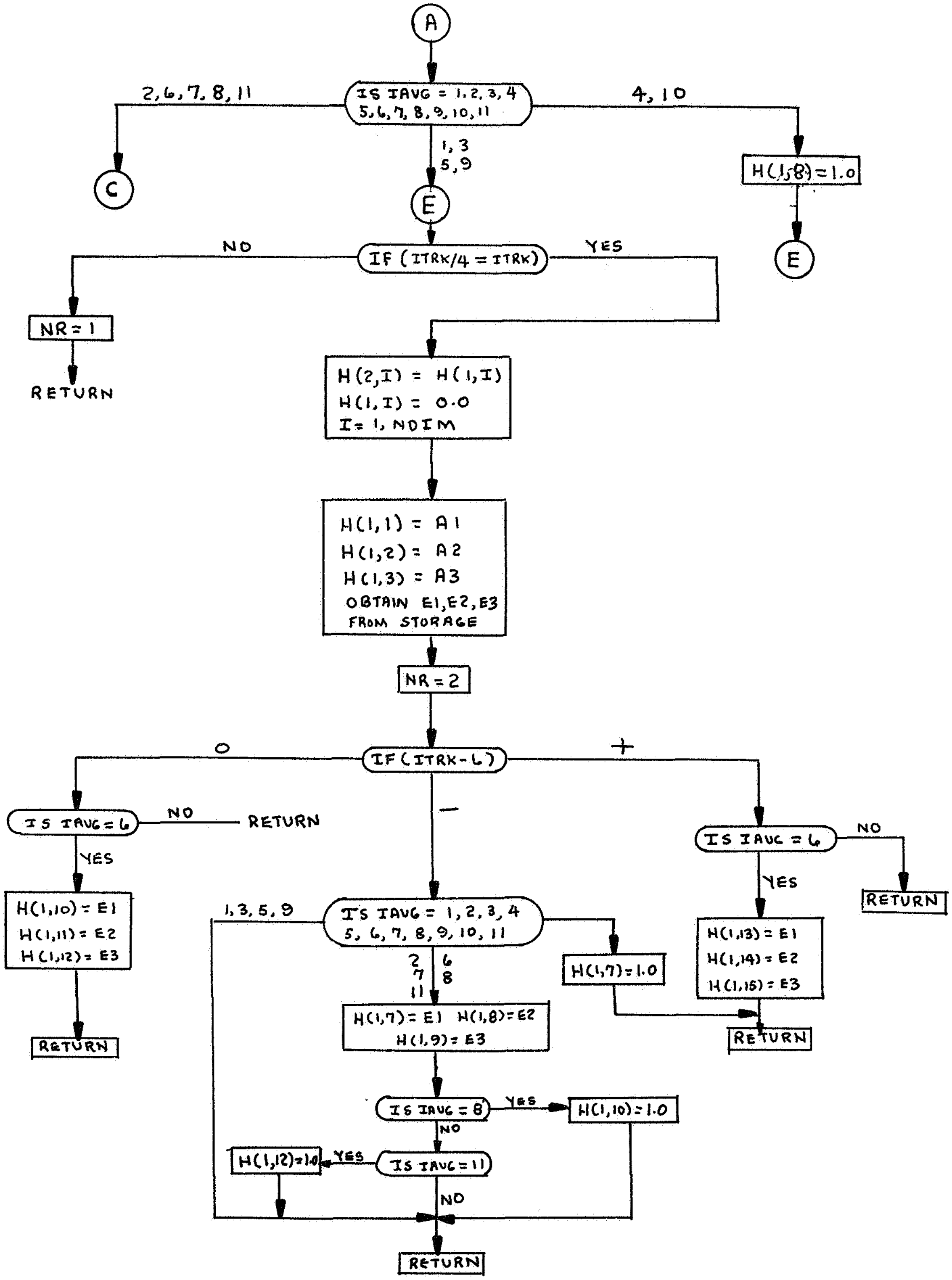
$$H = \begin{bmatrix} A_1 & A_2 & A_3 & 0 & 0 & 0 \\ B_1 & B_2 & B_3 & A_1 & A_2 & A_3 \end{bmatrix}$$

$$X_S = R_S \cos\Theta \cos[\phi + \omega(T-UNINT)]$$
$$Y_S = R_S \cos\epsilon \cos\Theta \sin[\phi + \omega(T-UNINT)] + R_S \sin\epsilon \sin\Theta$$
$$Z_S = -R_S \sin\epsilon \cos\Theta \sin[\phi + \omega(T-UNINT)] + R_S \cos\epsilon \sin\Theta$$
$$\dot{X}_S = -\omega R_S \cos\Theta \sin[\phi + \omega(T-UNINT)]$$
$$\dot{Y}_S = \omega R_S \cos\epsilon \cos\Theta \cos[\phi + \omega(T-UNINT)]$$
$$\dot{Z}_S = -\omega R_S \sin\epsilon \cos\Theta \cos[\phi + \omega(T-UNINT)]$$

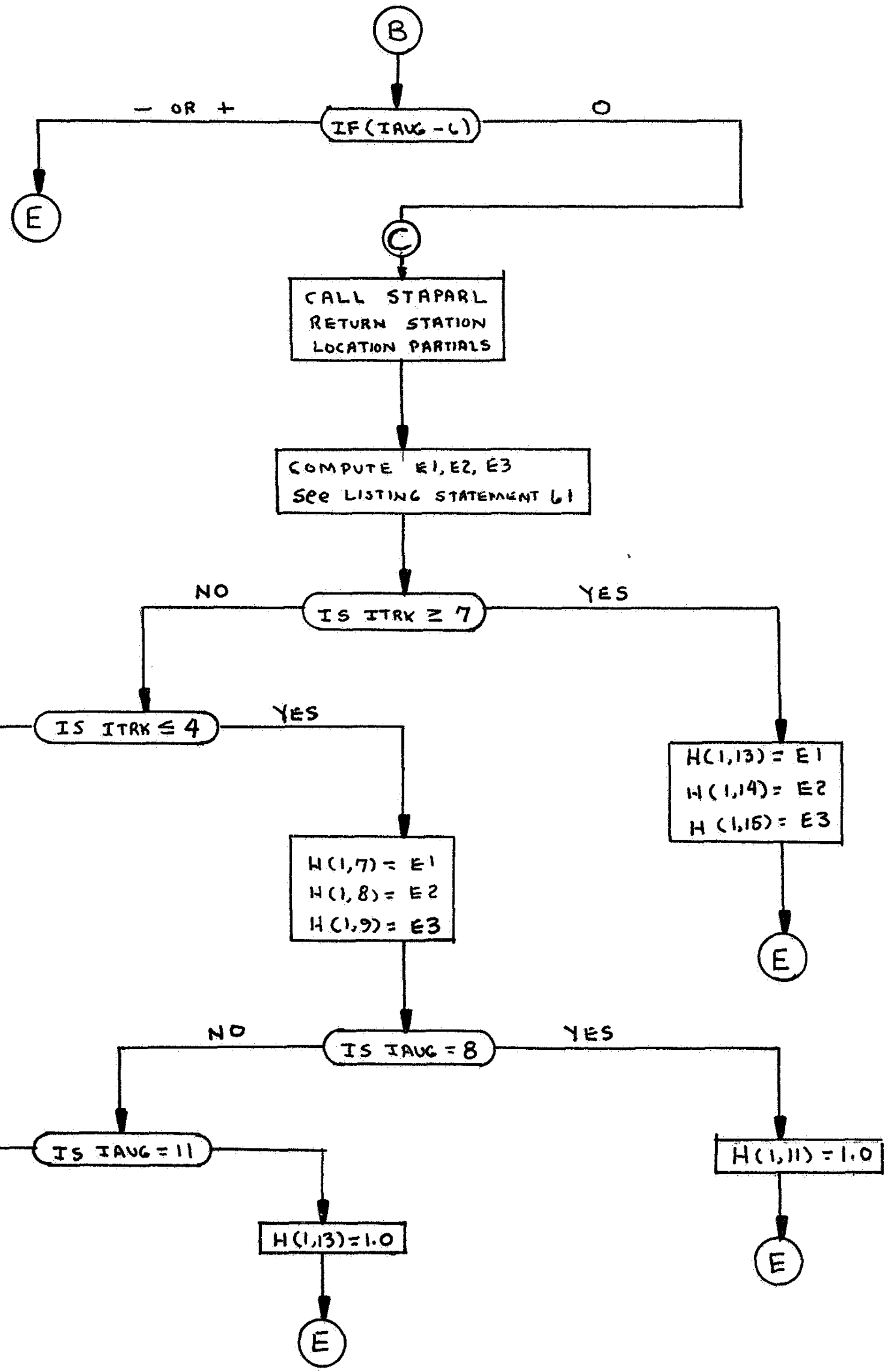
$$\vec{X} = \vec{X}_V - \vec{X}_E - \vec{X}_S$$
$$R = (X^2 + Y^2 + Z^2)^{1/2}$$
$$\dot{R} = (X\dot{X} + Y\dot{Y} + Z\dot{Z})/R$$

$$A_1 = \frac{X}{R} \quad A_2 = \frac{Y}{R} \quad A_3 = \frac{Z}{R}$$
$$B_1 = \frac{\dot{X}}{R} - \frac{X\dot{R}}{R^2} \quad B_2 = \frac{\dot{Y}}{R} - \frac{Y\dot{R}}{R^2}$$
$$B_3 = \frac{\dot{Z}}{R} - \frac{Z\dot{R}}{R^2}$$

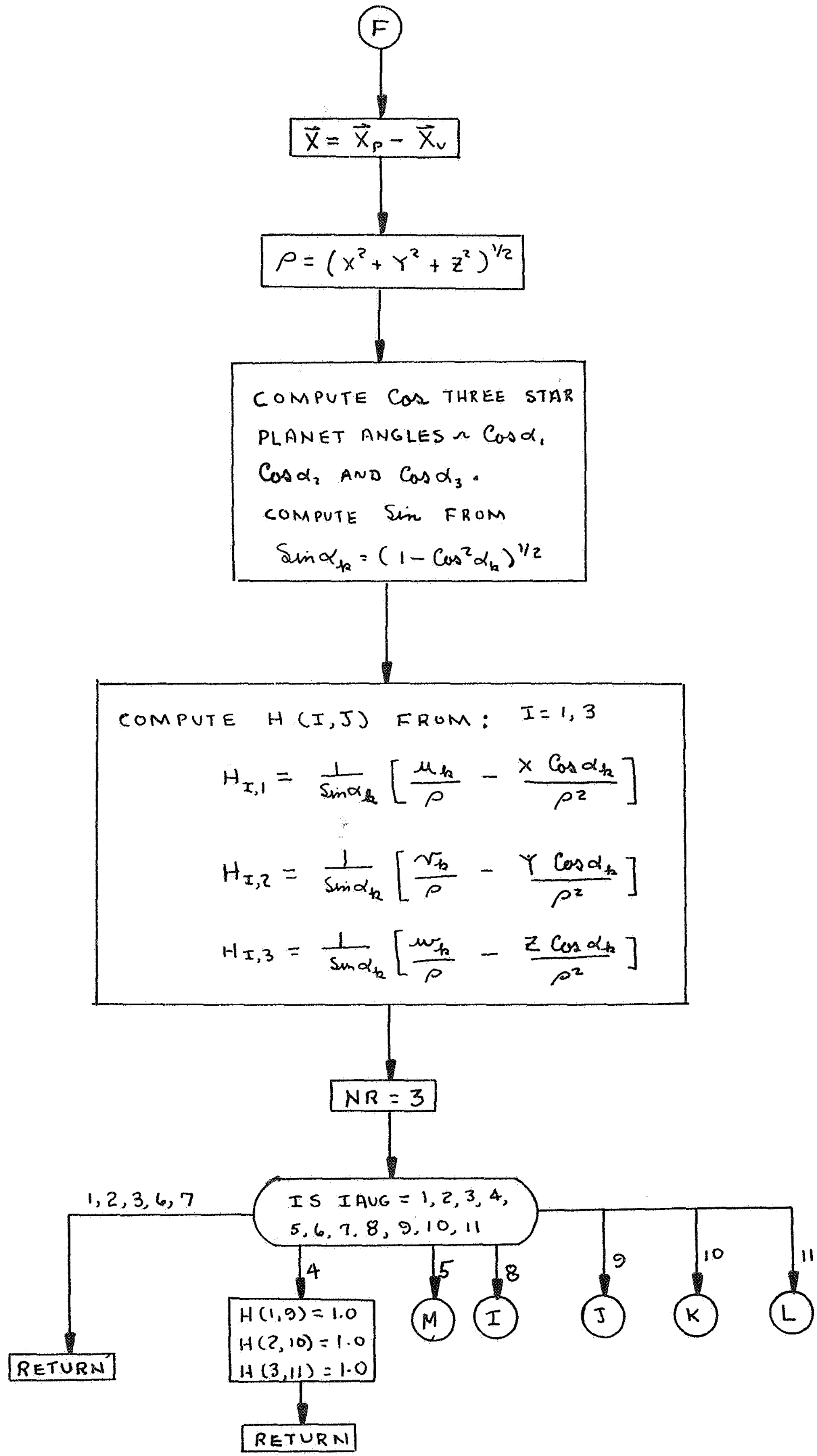
Computational logic (continued):



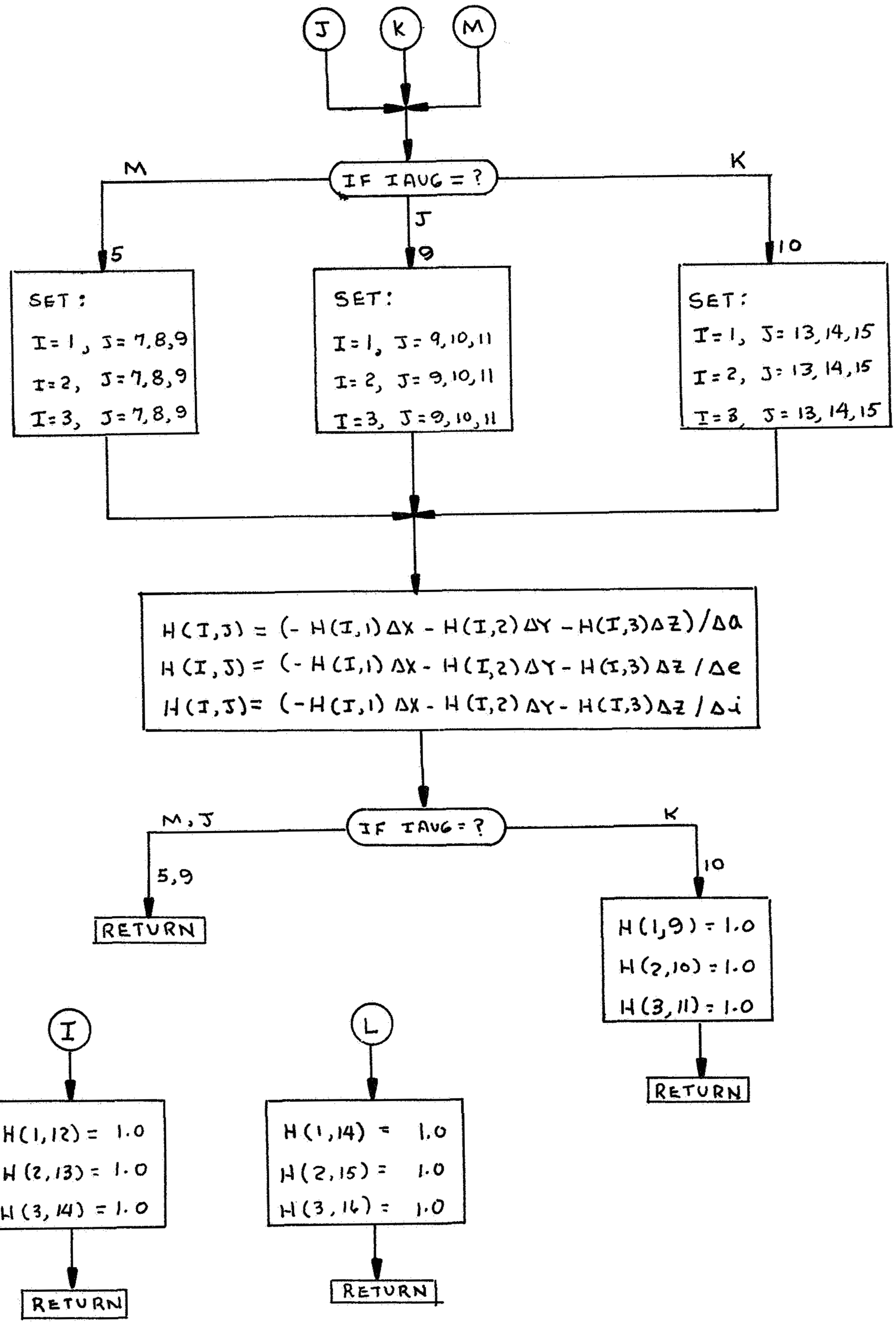
Computational logic (continued):



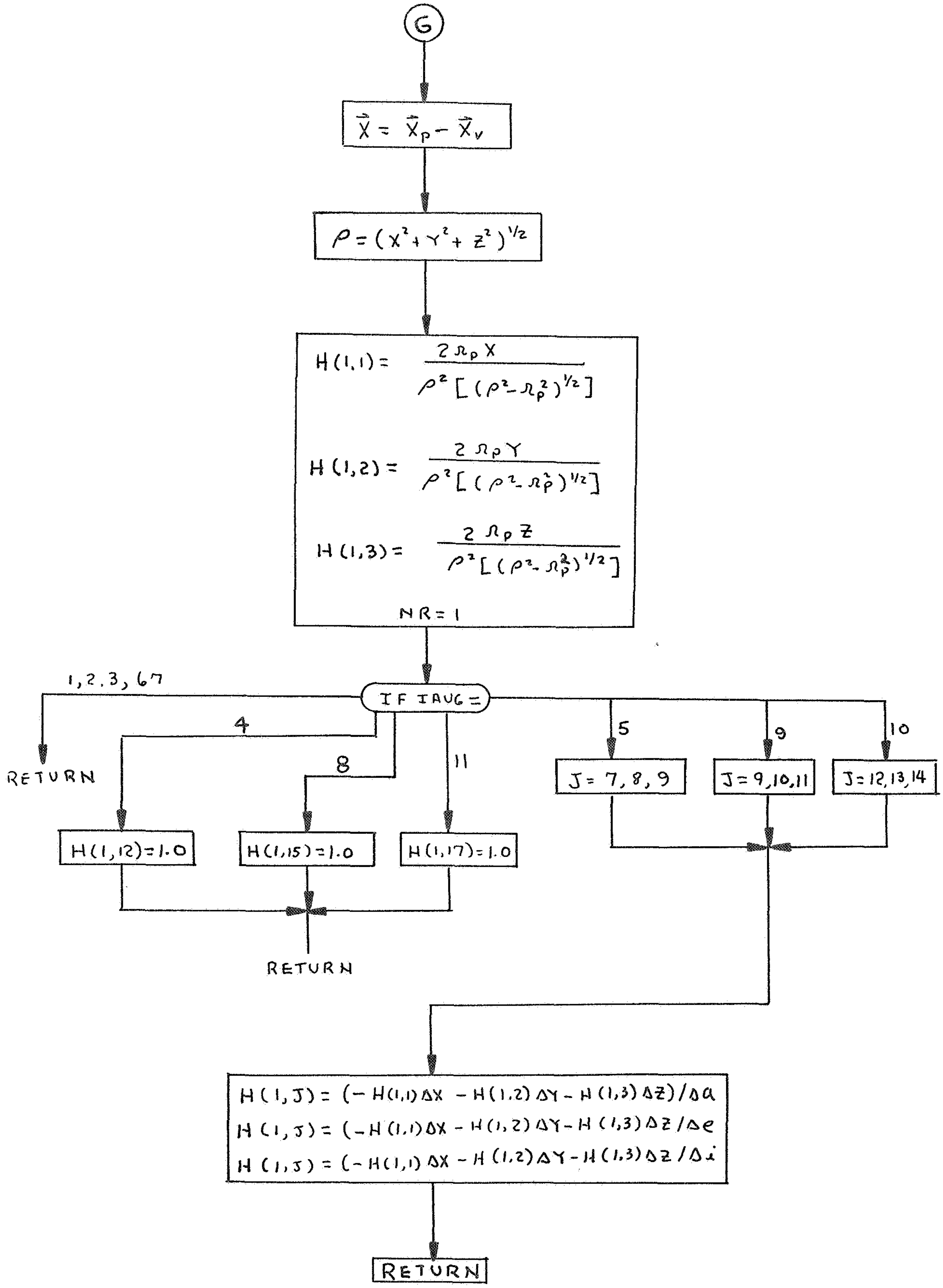
Computational logic (continued):



Computational logic (continued):



Computational logic (concluded):



61. Subroutine TRANS

Purpose: Three options are available with this subroutine:

1) Convert from geocentric equatorial rectangular coordinates to geocentric ecliptic coordinates;
2) Convert from geocentric equatorial coordinates to heliocentric ecliptic coordinates;
3) Convert from geocentric ecliptic coordinates to heliocentric ecliptic coordinates.

Calling sequence: CALL TRANS (ICODE, X, Y, Z, VX, VY, VZ, XE, YE, ZE, VXE, VYE, VZE, EPS, ICODE2).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	ICODE		An internal code that determines if option 1 or 2 above will be exercised.
I/O	X	x	X-component of the vehicle.
I/O	Y	y	Y-component of the vehicle.
I/O	Z	z	Z-component of the vehicle.
I/O	VX	$\dot{x}$	X-velocity component of the vehicle.
I/O	VY	$\dot{y}$	Y-velocity component of the vehicle.
I/O	VZ	$\dot{z}$	Z-velocity component of the vehicle.
I	XE	x_E	X-component of Earth in heliocentric ecliptic.
I	YE	y_E	Y-component of Earth.
I	ZE	z_E	Z-component of Earth.
I	VXE	$\dot{x}_E$	X-velocity component of Earth in heliocentric ecliptic.
I	VYE	$\dot{y}_E$	Y-velocity component of Earth.
I	VZE	$\dot{z}_E$	Z-velocity component of Earth.
I	EPS	ε	Obliquity of Earth.
I	ICODE2		An internal code that determines if option 3 above is to be exercised.

Subprograms required: None.

Approximate storage required (octal): 110.

Discussion: The position and velocity components of the vehicle are input to this routine in the desired coordinate system as stated above. After conversion is made to a new coordinate system, the position and velocity of the vehicle will be returned in the same locations. A full description of the input codes and equations used follows:

Option 1: ICODE = 1, ICODE2 = 1.

$$x = x \qquad \dot{x} = \dot{x}$$

$$y = y \cos \varepsilon + z \sin \varepsilon \qquad \dot{y} = \dot{y} \cos \varepsilon + \dot{z} \sin \varepsilon$$

$$z = -y \sin \varepsilon + z \cos \varepsilon \qquad \dot{z} = -\dot{y} \sin \varepsilon + \dot{z} \cos \varepsilon$$

Option 2: ICODE = 2, ICODE2 = 1.

The same procedure as above is used to convert from geocentric equatorial to geocentric ecliptic.

$$x = x + x_E \qquad \dot{x} = \dot{x} + \dot{x}_E$$

$$y = y + y_E \qquad \dot{y} = \dot{y} + \dot{y}_E$$

$$z = z + z_E \qquad \dot{z} = \dot{z} + \dot{z}_E$$

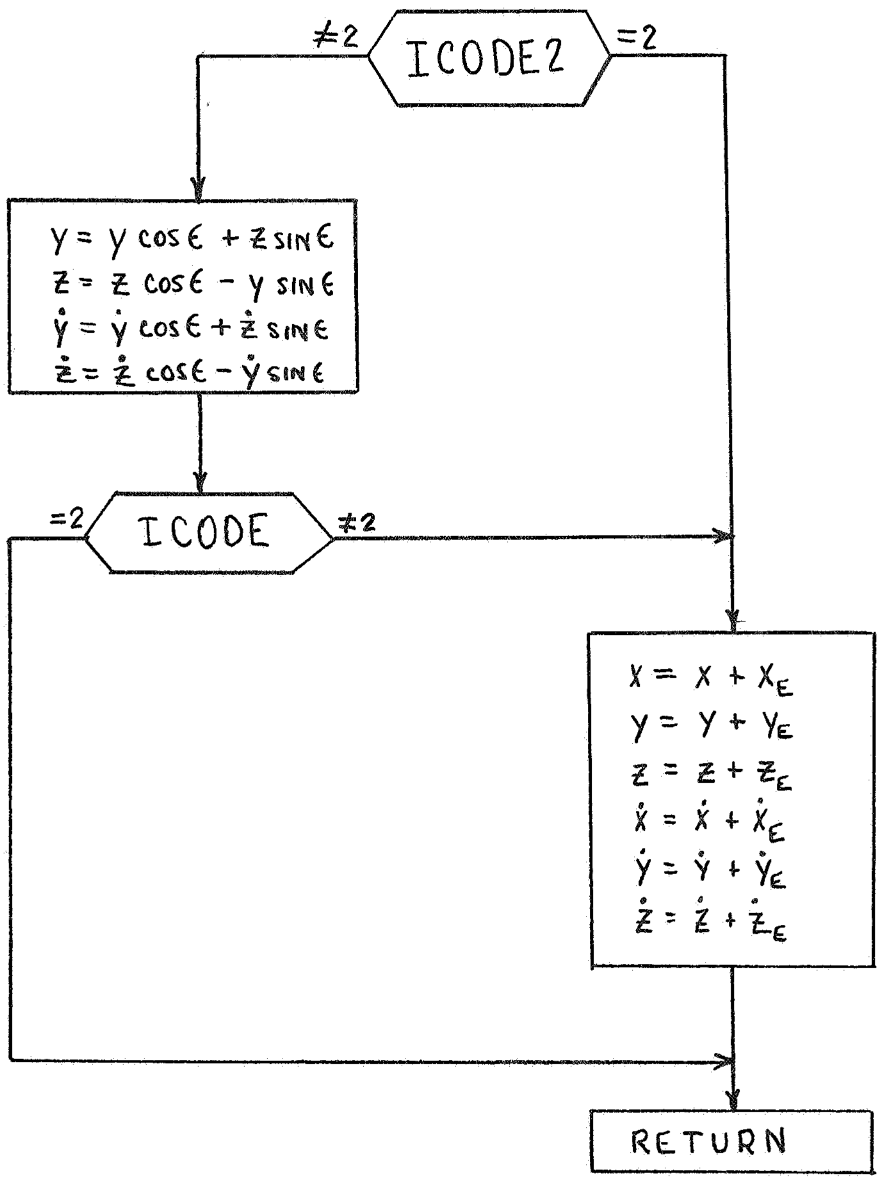
Option 3: ICODE = 2, ICODE2 = 2.

$$x = x + x_E \qquad \dot{x} = \dot{x} = \dot{x}_E$$

$$y = y + y_E \qquad \dot{y} = \dot{y} + \dot{y}_E$$

$$z = z + z_E \qquad \dot{z} = \dot{z} + \dot{z}_E$$

Computational logic:



62. Subroutine VARADA

Purpose: The variation matrix is built up by numerical differencing for the three variable B-plane guidance policy in the guidance event of the error analysis mode.

Calling sequence: CALL VARADA (RI, XSIP, XSIV, TEVN, TSI, ADA, B, BDT, BDR).

Input/output:

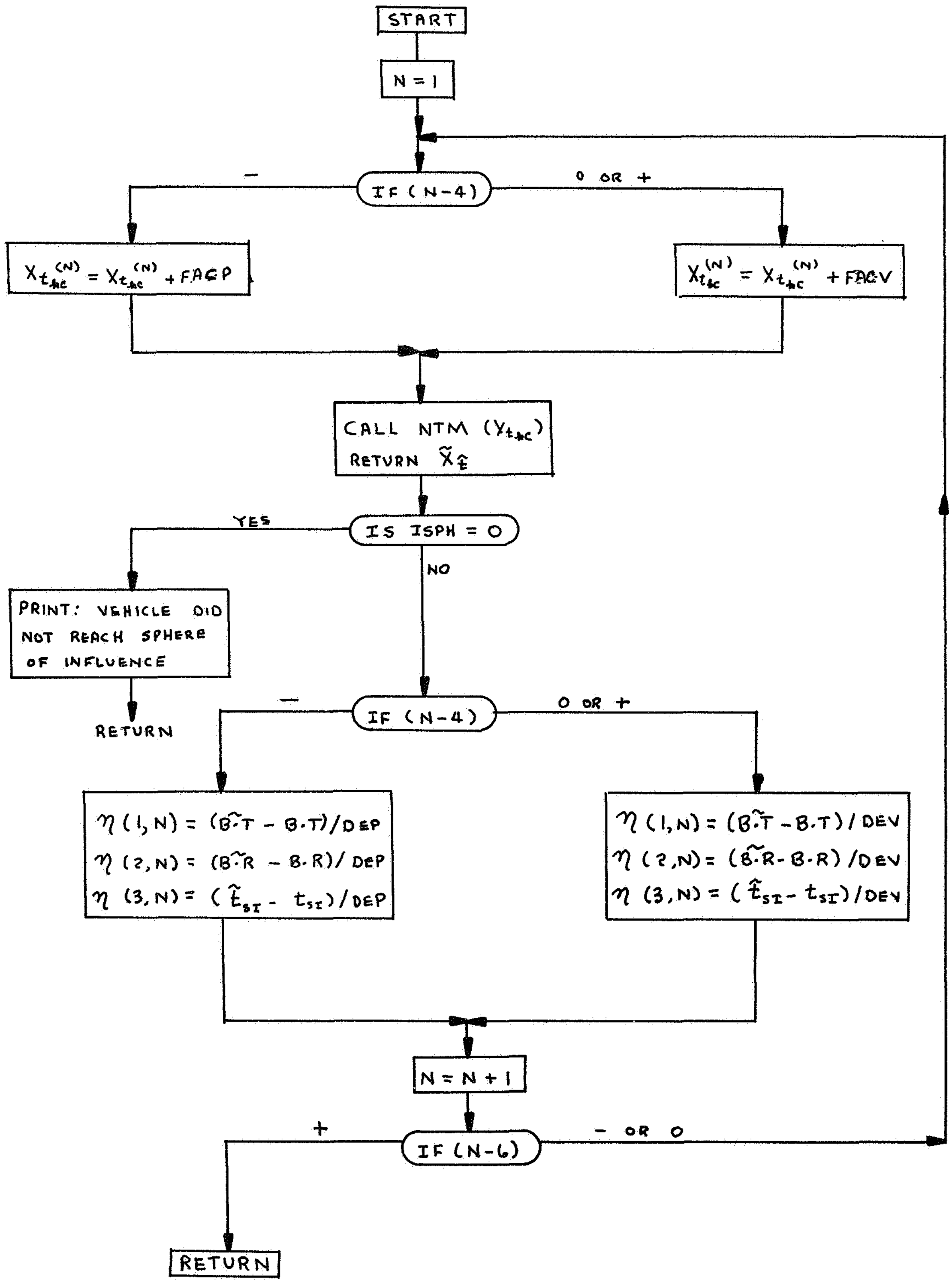
I/O	Fortran name	Math symbol	Definition
I	RI(6)	$\bar{X}$	Position and velocity of the vehicle at the time of the guidance event.
I	XSIP(3)	$\bar{r}_{SI}$	Position of the vehicle at the sphere of influence on the nominal trajectory.
I	XSIV(3)	$\bar{v}_{SI}$	Velocity of the vehicle at the sphere of influence on the nominal trajectory.
I	TEVN	t_{ev}	Trajectory time of the sphere of influence.
I	TSI	t_{SI}	Trajectory time at which the vehicle reached the sphere of influence on the nominal trajectory.
O	ADA(3,6)	η	Variation matrix.
I	B		B of the nominal trajectory.
I	BDT		B·T of the nominal trajectory.
I	BDR		B·R of the nominal trajectory.

Subprograms required: NTM.

Approximate storage required (octal): 320.

Discussion: The variation matrix, η, is generated using the numerical differencing technique. Each component of the state vector of the vehicle at the time of the guidance event is altered in its turn and changes in $B \cdot T$, $B \cdot R$ and the time at which the vehicle reaches the sphere of influence are noted. Then the usual method is applied to obtain the 3 x 6 variation matrix.

Computational logic:



63. Subroutine VARSIM

Purpose: The variation matrix, η, is computed for the three-variable B-plane guidance policy in the guidance event of the simulation mode.

Calling sequence: CALL VARSIM (RI1, TEVN, TSI, ADA).

Input/output:

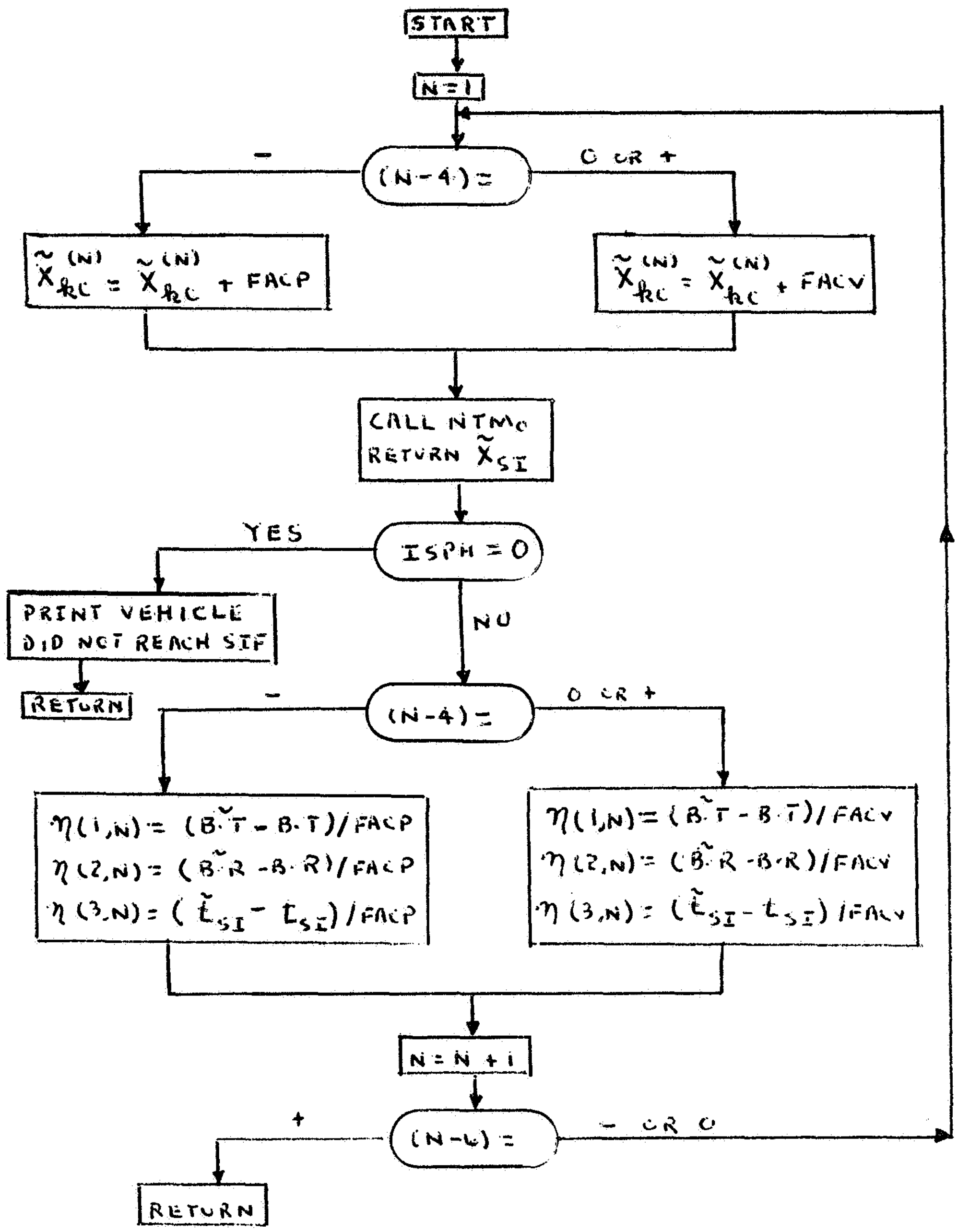
I/O	Fortran name	Math symbol	Definition
I	RI1(6)	$\tilde{X}$	Position and velocity of the vehicle on the most recent nominal trajectory at the time of the guidance event.
I	TEVN	t_{ev}	Trajectory time of the guidance event.
I	TSI	t_{SI}	Trajectory time at which the vehicle reached the sphere of influence on the most recent nominal trajectory.
O	ADA(3,6)	η	Variation matrix.

Subprograms required: NTM.

Approximate storage required (octal): 270.

Discussion: The variation matrix is generated for the three-variable B-plane in this routine. The numerical differencing technique is used. By altering independently each component of the position and velocity of the vehicle at the time of the guidance event, the changes in $B \cdot T$, $B \cdot R$, and t_{SI} are noted. The usual method is then applied to obtain the 3 x 6 variation matrix.

Computational logic:



64. Subroutine VECTOR

Purpose: This subroutine calculates the vector orbital elements k, e, computes the spacecraft final position on the orbit to accurately approximate the desired time interval, and finally computes the conic section time of flight.

Calling sequence: CALL VECTOR.

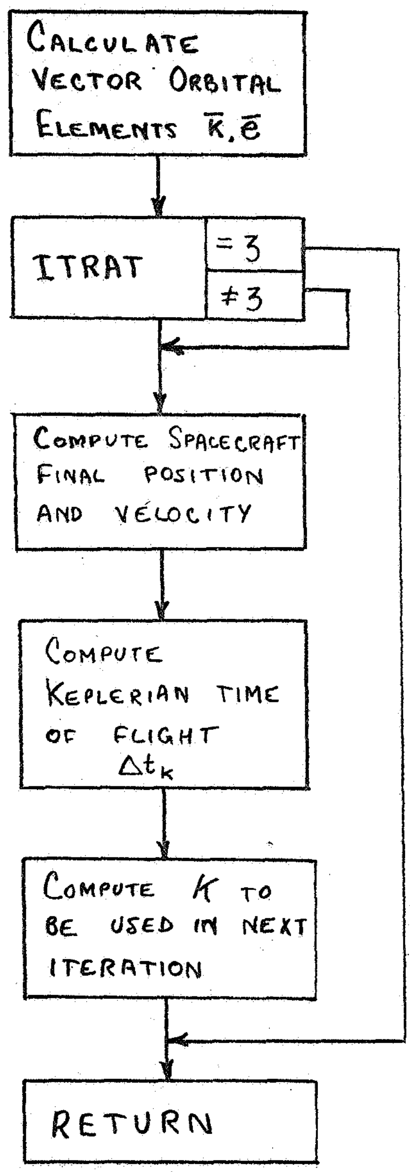
Input/output: All communication with VECTOR is accomplished through COMMON statements.

Subprograms required: SPACE.

Approximate storage required (octal): 570.

Discussion: For a complete formulation of the techniques used in this routine refer to the Analytical Manual (Volume II) of this report.

Computational logic:



65. Subroutine VMASS

Purpose: This subroutine determines the virtual mass data needed in the trajectory analysis. The virtual mass position, velocity, magnitude, and magnitude rate are calculated.

Calling sequence: CALL VMASS.

Input/output: All communication with this routine is through COMMON statements.

Subprograms required: None.

Approximate storage required (octal): 330.

Discussion: The formulas used in this subroutine are discussed in the Analytical Manual, Volume II.

66. Subroutine VMP

Purpose: This subroutine is responsible for the logic involved in generating a virtual mass trajectory.

Calling sequence: CALL VMP (RS, ACC, D1, TRTM, DELTM, RSF, ISP2).

Input/output:

I/O	Fortran name	Math symbol	Definition
I	RS(6)	$\bar{X}_i$	Position and velocity of vehicle at initial time.
I	ACC	$\Delta r/r$	Accuracy used.
I	D1	d	Julian date, epoch January 0, 1900, of initial trajectory time.
I	TRTM	t_o	Initial trajectory time.
I	DELTM	Δt	Time interval over which trajectory is to be generated.
0	RSF(6)	$\bar{X}_f$	Position and velocity of vehicle at final time.
I	ISP2		A code that determines if the trajectory is to stop at sphere of influence.

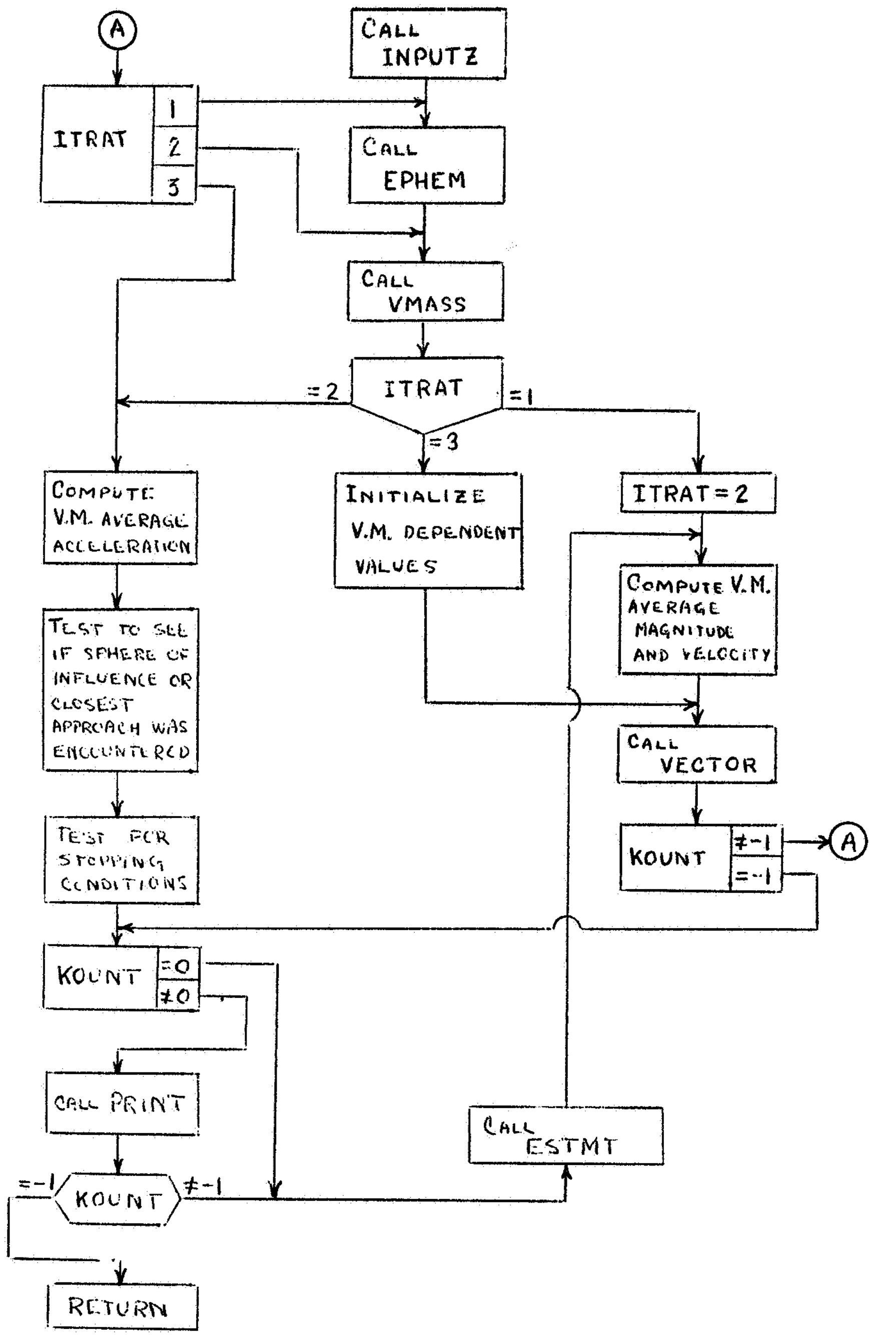
Subprograms required: ACTB, EPHEM, ESTMT, INPUTZ, NEWPGE, ORB, PRINT, SPACE, TIME, VECTOR, VMASS.

Approximate storage required (octal): 1730.

Discussion: This subroutine handles the general flow of the virtual mass program. In addition it checks to see if the vehicle has encountered closest approach or the sphere of influence of the target planet. If closest approach or the sphere of influence is reached during the time increment over which the trajectory is being computed, the program will not check for a second closest approach or a second encounter with the sphere of influence.

Also, only one target planet may be specified for a given run. Thus, it is not possible to check for closest approach or sphere of influence for a second target planet.

Computation logic:



VI. VARIABLE LIST

IN THIS CHAPTER EACH VARIABLE USED IN THE STEAP PROGRAM IS LISTED AND DESCRIBED. AS A GREAT NUMBER OF THESE VARIABLES APPEAR IN COMMON BLOCKS AND IN ORDER TO PREVENT REPETITION, ALL COMMON BLOCKS ARE TREATED SEPARATELY. SUBSEQUENTLY, ONLY THE NAME OF THE COMMON STATEMENT IS LISTED IN THE SUBROUTINE.

COMMON BLOCKS --

BLK

T	--	TRAJECTORY TIME IN DAYS
PMASS(11)	--	GRAVITATIONAL CONSTANTS OF PLANETS IN A.U.**3/DAY**2
CN(80)	--	CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE FIRST FIVE PLANETS
ST(50)	--	CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE LAST FOUR PLANETS
EMN(15)	--	THE CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE MOON
SMJR(18)	--	CONSTANTS USED TO CALCULATE THE SEMI-MAJOR AXES OF THE PLANETS
RADIUS(11)	--	THE RADIUS OF A GIVEN PLANET IN A.U.
RMASS(11)	--	THE RELATIVE GRAVITATIONAL CONSTANT OF A STATED PLANET WITH RESPECT TO THE SUN
NO(11)	--	AN ARRAY OF PLANET CODES BEING USED TO GENERATE THE VIRTUAL MASS TRAJECTORY
ELMNT(80)	--	CONTAINS THE ORBITAL ELEMENTS OF THE PLANETS
SPHERE(11)	--	THE SPHERES OF INFLUENCE OF THE PLANETS IN A.U.
XP(6)	--	THE POSITION AND VELOCITY OF A PLANET IN HELIOCENTRIC ECLIPTIC COORDINATES

COM

V(16,7)	--	AN ARRAY WHICH STORES PERTINANT VECTORS USED IN THE CALCULATION OF THE VIRTUAL MASS TRAJECTORY
F(44,4)	--	CONTAINS THE POSITIONS AND VELOCITIES OF THE PLANETS AT A SPECIFIED TIME PLUS THE POSITIONS AND VELOCITIES OF THE SPACECRAFT RELATIVE TO THE PLANETS
PI	--	THE VALUE OF THE MATHEMATICAL CONSTANT PI
RAD	--	THE NUMBER OF DEGREES PER RADIAN
ITRAT	--	IN INTERNAL CODE USED TO DETERMINE HOW MANY ITERATIONS HAVE BEEN ACCOMPLISHED IN THE VIRTUAL MASS PROCEDURE
KOUNT	--	A CODE WHICH SPECIFIES WHETHER PRINT-OUT IS TO OCCUR AFTER THIS TIME INCREMENT
INCMNT	--	NUMBER OF INCREMENTS USED
INCPR	--	SPECIFIES AFTER HOW MANY TIME INCREMENTS PRINT-OUT IS TO OCCUR

*

INC	--	DETERMINE WHETHER THE ABOVE OPION IS TO BE USED
IPR	--	A CODE WHICH DETERMINES IF PRINT-OUT IS TO OCCUR AFTER A SPECIFIED NUMBER OF DAYS
NBODYI	--	NUMBER OF BODIES CONSIDERED IN VIRTUAL MASS TRAJECTORY
NBODY	--	BASED ON ABOVE VALUE--EQUAL TO 4*NBODYI-3
IPRT(4)	--	SPECIFIES PRINT OPTIONS (IN STEAP TRAJECTORY THIS OPTION IS OMITTED. WHEN PRINT-OUT OCCURS ALL SECTIONS ARE AUTOMATICALLY PRINTED)
KL	--	PROBLEM NUMBER
IPG	--	PAGE NUMBER
LINCT	--	LINE COUNT
LINPGE	--	LINES PER PAGE

CONST

OMEGA	--	EARTH'S ROTATION RATE
FPS	--	OBLIQUITY OF EARTH
NST	--	NUMBER OF STATIONS TO BE USED (MAXIMUM 3)
SAL(3)	--	ALTITUDES OF STATIONS
SLAT(3)	--	LATITUDES OF STATIONS
SLON(3)	--	LONGITUDES OF STATIONS
DNCN(3)	--	CONSTANTS FROM WHICH DYNAMIC NOISE IS COMPUTED
MNCN(12)	--	MEASUREMENT NOISE CONSTANTS

CONST2

U1	--	DIRECTION COSINE OF STAR 1
U2	--	DIRECTION COSINE OF STAR 2
U3	--	DIRECTION COSINE OF STAR 3
V1	--	DIRECTION COSINE OF STAR 1
V2	--	DIRECTION COSINE OF STAR 2
V3	--	DIRECTION COSINE OF STAR 3
W1	--	DIRECTION COSINE OF STAR 1
W2	--	DIRECTION COSINE OF STAR 2
W3	--	DIRECTION COSINE OF STAR 3
FOP	--	OFF-DIAGONAL ANNIHILATION VALUE FOR POSITION EIGENVALUES
FOV	--	OFF-DIAGONAL ANNIHILATION VALUE FOR VELOCITY EIGENVALUES

CONST3

DELXA	--	THE AMOUNT OF CHANGE IN X-COMPONENT DUE TO CHANGE IN SEMI-MAJOR AXIS OF TARGET PLANET
DELYA	--	THE AMOUNT OF CHANGE IN Y-COMPONENT DUE TO CHANGE IN SEMI-MAJOR AXIS OF TARGET PLANET
DELZA	--	THE AMOUNT OF CHANGE IN Z-COMPONENT DUE TO CHANGE IN SEMI-MAJOR AXIS OF TARGET PLANET
DELXE	--	THE AMOUNT OF CHANGE IN X-COMPONENT DUE TO CHANGE IN ECCENTRICITY OF TARGET PLANET
DELYE	--	THE AMOUNT OF CHANGE IN Y-COMPONENT DUE TO CHANGE IN ECCENTRICITY OF TARGET PLANET
DELZE	--	THE AMOUNT OF CHANGE IN Z-COMPONENT DUE TO CHANGE IN ECCENTRICITY OF TARGET PLANET
DELXI	--	THE AMOUNT OF CHANGE IN X-COMPONENT DUE TO CHANGE IN INCLINATION OF TARGET PLANET

*

DELYI	--	THE AMOUNT OF CHANGE IN Y-COMPONENT DUE TO CHANGE IN INCLINATION OF TARGET PLANET
DELZI	--	THE AMOUNT OF CHANGE IN Z-COMPONENT DUE TO CHANGE IN INCLINATION OF TARGET PLANET
DELAXS	--	CHANGE IN SEMI-MAJOR AXIS OF TARGET PLANET USED TO GENERATE STATE TRANSITION MATRIX
DELECC	--	CHANGE IN ECCENTRICITY OF TARGET PLANET USED TO GENERATE STATE TRANSITION MATRIX
DELICL	--	CHANGE IN INCLINATION OF TARGET PLANET USED TO GENERATE STATE TRANSITION MATRIX
DELMUS	--	CHANGE IN GRAVITATIONAL CONSTANT OF SUN USED TO GENERATE STATE TRANSITION MATRIX
DELMUP	--	CHANGE IN GRAVITATIONAL CONSTANT OF TARGET PLANET USED TO GENERATE STATE TRANSITION MATRIX

EVENT

NEV	--	NUMBER OF EVENTS
TEV(50)	--	TIMES OF EVENTS
IEVNT(50)	--	CODES OF EVENTS
IHYP1	--	HYPERELLIPSOID CODE USED TO DETERMINE IF K=1, K=3, OR BOTH
IEIG	--	CODE USED TO DECIDE IF BOTH POSITION AND VELOCITY EIGENVECTORS ARE REQUESTED
TPT2(20)	--	PREDICTION TIMES
ICDT3(20)	--	CODES WHICH DETERMINE WHICH GUIDANCE POLICIES ARE BEING USED
NPE	--	NUMBER OF PREDICTION EVENTS HAVING OCCURRED
NGE	--	NUMBER OF GUIDANCE EVENTS HAVING OCCURRED
IPOL	--	CODE WHICH DETERMINES IF FIXED-TIME-OF-ARRIVAL GUIDANCE EVENT HAS OCCURED
IIPOL	--	CODE WHICH DETERMINES IF EITHER TWO-VARIABLE OR THREE-VARIABLE B-PLANE GUIDANCE POLICY HAS OCCURRED
ICDQ3(20)	--	ARRAY OF CODES WHICH DETERMINE WHICH EXECUTION POLICIES ARE TO BE USED IN GUIDANCE EVENTS
SIGRES	--	VARIANCE OF RESOLUTION ERROR
SIGPRO	--	VARIANCE OF PROPORTIONALITY ERROR
SIGALP	--	VARIANCE OF ERROR IN POINTING ANGLE 1
SIGBET	--	VARIANCE OF ERROR IN POINTING ANGLE 2
NEV1	--	TOTAL NUMBER OF EIGENVECTOR EVENTS
NEV2	--	TOTAL NUMBER OF PREDICTION EVENTS
NEV3	--	TOTAL NUMBER OF GUIDANCE EVENTS
NEV4	--	TOTAL NUMBER OF QUASI-LINEAR FILTERING EVENTS
NQE	--	QUASI-LINEAR FILTERING EVENTS HAVING OCCURRED

GUI

PG(17,17)	--	COVARIANCE MATRIX AT TIME OF LAST GUIDANCE EVENT
XG(6)	--	STATE VECTOR AT TIME OF LAST GUIDANCE EVENT
TG	--	TIME OF LAST GUIDANCE EVENT
EM(2,6)	--	PARTIAL OF B.T AND B.R WITH RESPECT TO STATE VECTOR

*

```
MEAS
  TMN(1000)    --  TIMES OF MEASUREMENTS
  MCODE(1000)--  ARRAY OF MEASUREMENT CODES
  NMN          --  TOTAL NUMBER OF MEASUREMENTS
  MCNTR        --  NUMBER OF MEASUREMENTS HAVING OCCURRED
MISC
  ACC          --  ACCURACY FIGURE USED IN VIRTUAL MASS PROGRAM
  IDNF         --  DYNAMIC NOISE FLAG
  ICOOR        --  STATE VECTOR CODE WHICH DETERMINES IN WHICH
                   COORDINATE SYSTEM THE VECTOR IS READ IN
  ITR          --  MODE FLAG
  IMNF         --  MEASUREMENT NOISE FLAG
  FACP         --  POSITION FACTOR USED IN NUMERICAL DIFFERENCING
  FACV         --  VELOCITY FACTOR USED IN NUMERICAL DIFFERENCING
  ISP2         --  SPHERE OF INFLUENCE FLAG
  BIA(12)      --  MEASUREMENT BIASES
  IPGN         --  PAGE NUMBER

NAME
  MDNM(4,2)    --  MODE NAME
  EVNM(4)      --  EVENT NAME
  MNNAME(12,3--  MEASUREMENT NAME
  CMPNM(11,17--  COMPONENT NAME

PRT
  MONTH(12)    --  NAMES OF MONTHS
  PLANET(11)   --  NAMES OF PLANETS

SIMCNT
  DMUSB        --  BIAS IN GRAVITATIONAL CONSTANT OF SUN
  DMUPB        --  BIAS IN GRAVITATIONAL CONSTANT OF TARGET
                   PLANET
  DAB          --  BIAS IN SEMI-MAJOR AXIS OF TARGET PLANET
  DEB          --  BIAS IN ECCENTRICITY OF TARGET PLANET
  DIB          --  BIAS IN INCLINATION OF TARGET PLANET
  TTIM1        --  FIRST TIME USED FOR UNMODELLED ACCELERATION
  TTIM2        --  SECOND TIME USED FOR UNMODELLED ACCELERATION
  UNMAC(3,3)   --  UNMODELLED ACCELERATION
  SLB(9)       --  BIASES IN STATION LOCATION CONSTANTS
  AVARM(12)    --  VARIANCE OF ACTUAL MEASUREMENT NOISE
  IAMNF        --  ACTUAL MEASUREMENT NOISE FLAG
  ARES(20)     --  ACTUAL RESOLUTION ERROR
  APRO(20)     --  ACTUAL PROPORTIONALITY ERROR
  AALP(20)     --  ACTUAL ERROR IN POINTING ANGLE 1
  ABET(20)     --  ACTUAL ERROR IN POINTING ANGLE 2

SIM1
  XI1(17)      --  INITIAL STATE VECTOR OF MOST RECENT NOMINAL
                   TRAJECTORY
  XF1(17)      --  FINAL STATE VECTOR OF MOST RECENT NOMINAL
                   TRAJECTORY
  ADEVX(17)    --  ACTUAL DEVIATION IN THE STATE VECTOR
  EDEVX(17)    --  ESTIMATED DEVIATION IN THE STATE VECTOR
  W(17)        --  ACTUAL DYNAMIC NOISE
```

*

```
  Z(17)        --  ACTUAL STATE VECTOR
  ANOIS(17)    --  ACTUAL WHITE NOISE
  RES(4)       --  RESIDUAL
  EY(4)        --  ESTIMATED MEASUREMENT
  AY(4)        --  ACTUAL MEASUREMENT
  AR(4,4)      --  ACTUAL MEASUREMENT NOISE
  ZI(17)       --  INITIAL ACTUAL STATE VECTOR
  ADEVXB(17)   --  ACTUAL DEVIATION IN STATE VECTOR AT BEGINNING
                   OF TRAJECTORY

SIM2
  NB1(11)      --  ARRAY OF PLANET CODES IN ACTUAL TRAJECTORY
  ACC1         --  ACCURACY USED IN ACTUAL TRAJECTORY
  NBOD1        --  NUMBER OF BODIES IN ACTUAL TRAJECTORY

STM
  P(17,17)     --  COVARIANCE MATRIX
  PSI(17,17)   --  STATE TRANSITION MATRIX
  Q(17,17)     --  DYNAMIC NOISE MATRIX
  H(4,17)      --  OBSERVATION MATRIX
  R(4,4)       --  MEASUREMENT NOISE MATRIX
  AK(17,4)     --  K MATRIX
  PB(17,17)    --  COVARIANCE MATRIX AT BEGINNING OF TRAJECTORY
  PSIP(17,17)--    COVARIANCE MATRIX BEFORE THE MEASUREMENT
  HPHR(17,17)--    H*P*H-TRANSPOSE + R

STVEC
  XI(17)       --  INITIAL STATE VECTOR OF ORIGINAL NOMINAL
  XF(17)       --  FINAL STATE VECTOR OF ORIGINAL NOMINAL
  NDIM         --  DIMENSION OF STATE VECTOR
  IAUG         --  AUGMENTATION CODE
  XB(17)       --  STATE VECTOR OF ORIGINAL NOMINAL AT BEGINNING
                   OF TRAJECTORY

  TIM
  TRTM1        --  INITIAL TRAJECTORY TIME
  DELTM        --  TIME INCREMENT
  FNTM         --  FINAL TRAJECTORY TIME
  UN1VT        --  UNIVERSAL TIME
  TRTMB        --  TRAJECTORY TIME AT BEGINNING OF TRAJECTORY

TRJ
  ISOI1        --  SPHERE OF INFLUENCE CODE FOR ORIGINAL NOMINAL
  ISOI2        --  SPHERE OF INFLUENCE CODE FOR MOST RECENT
                   NOMINAL
  ISOI3        --  SPHERE OF INFLUENCE CODE FOR ACTUAL TRAJECTORY
  ICA1         --  CLOSEST APPROACH CODE FOR ORIGINAL NOMINAL
  ICA2         --  CLOSEST APPROACH CODE FOR MOST RECENT NOMINAL
  ICA3         --  CLOSEST APPROACH CODE FOR ACTUAL TRAJECTORY
  RCA1(6)      --  STATE AT CLOSEST APPROACH ON ORIGINAL NOMINAL
  RCA2(6)      --  STATE AT CLOSEST APPROACH ON MOST RECENT
                   NOMINAL
  RCA3(6)      --  STATE AT CLOSEST APPROACH ON ACTUAL TRAJECTORY
  RSOI1(3)     --  POSITION AT SPHERE OF INFLUENCE ON ORIGINAL
                   NOMINAL
```

*

```
RSOI2(3)    --  POSITION AT SPHERE OF INFLUENCE ON MOST RECENT
                NOMINAL
RSOI3(3)    --  POSITION AT SPHERE OF INFLUENCE ON ACTUAL
                TRAJECTORY
VSOI1(3)    --  VELOCITY AT SPHERE OF INFLUENCE ON ORIGINAL
                NOMINAL
VSOI2(3)    --  VELOCITY AT SPHERE OF INFLUENCE ON MOST RECENT
                NOMINAL
VSOI3(3)    --  VELOCITY AT SPHERE OF INFLUENCE ON ACTUAL
                TRAJECTORY
TCA1        --  TIME AT CLOSEST APPROACH OF ORIGINAL NOMINAL
TCA2        --  TIME AT CLOSEST APPROACH OF MOST RECENT
                NOMINAL
TCA3        --  TIME AT CLOSEST APPROACH OF ACTUAL TRAJECTORY
TSOI1       --  TIME AT SPHERE OF INFLUENCE OF ORIGINAL
                NOMINAL
TSOI2       --  TIME AT SPHERE OF INFLUENCE OF MOST RECENT
                NOMINAL
TSOI3       --  TIME AT SPHERE OF INFLUENCE OF ACTUAL
                TRAJECTORY
BSI1        --  B ON ORIGINAL NOMINAL
BSI2        --  B ON MOST RECENT NOMINAL
BSI3        --  B ON ACTUAL TRAJECTORY
BDTSI1      --  B DOT T ON ORIGINAL NOMINAL
BDTSI2      --  B DOT T ON MOST RECENT NOMINAL
BDTSI3      --  B DOT T ON ACTUAL TRAJECTORY
BDRSI1      --  B DOT R ON ORIGINAL NOMINAL
BDRSI2      --  B DOT R ON MOST RECENT NOMINAL
BDRSI3      --  B DOT R ON ACTUAL TRAJECTORY

TRAJCD
 NTMC       --  NOMINAL TRAJECTORY CODE
 ISTMC      --  STATE TRANSITION MATRIX CODE
 ISTM1      --  ALTERNATE STATE TRANSITION MATRIX CODE
 DTMAX      --  MAXIMUM TIME INCREMENT FOR WHICH ISTMC IS
                VALID
 NDACC      --  NUMERICAL DIFFERENCING ACCURACY CODE
 ACCND      --  ACCURACY USED IN NUMERICAL DIFFERENCING IF
                NDACC INDICATES

 VM
 NBOD       --  NUMBER OF BODIES USED IN VIRTUAL MASS PROGRAM
 NB(11)     --  CODES OF PLANETS
 NTP        --  CODE OF TARGET PLANET
 ALNGTH     --  LENGTH UNITS PER A.U.
 TM         --  TIME UNITS PER DAY
 DELTP      --  PRINT INCREMENTS (IN DAYS)
 INPR       --  PRINT INCREMENTS (IN INCREMENTS)
 IPROB      --  PROBLEM NUMBER
 RC(6)      --  STATE AT CLOSEST APPROACH
 DC         --  JULIAN DATE,EPOCH 1900,AT CLOSEST APPROACH
 RSI(3)     --  POSITION AT SPHERE OF INFLUENCE
 VSI(3)     --  VELOCITY AT SPHERE OF INFLUENCE
 DSI        --  JULIAN DATE, EPOCH 1900, AT SPHERE OF
                INFLUENCE
```

*

ISPH	--	SPHERE OF INFLUENCE CODE
RVS(6)	--	POSITION OF VEHICLE RELATIVE TO VIRTUAL MASS
VMU	--	GRAVITATIONAL CONSTANT OF VIRTUAL MASS
B	--	B AT SPHERE OF INFLUENCE
BDT	--	B DOT T
BDR	--	B DOT R
DELTH	--	INCREMENT IN TRUE ANOMALY USED
TIMINT	--	TOTAL TIME USED
INCMT	--	TOTAL INCREMENTS USED
IEPHEM	--	EPHEMERIS CODE
ICL	--	CLOSEST APPROACH CODE
IPRINT	--	PRINT CODE
RE(6)	--	POSITION AND VELOCITY OF EARTH
RTP(6)	--	POSITION AND VELOCITY OF TARGET PLANET
ICL2	--	CLOSEST APPROACH TERMINATION CODE

STEAP--MAIN PROGRAM

COMMON
- BLK
- COM
- CONST
- CONST2
- CONST3
- EVENT
- GUI
- MEAS
- MISC
- NAME
- PRT
- SIMCNT
- SIM1
- SIM2
- STM
- STVEC
- TIM
- TRAJCD
- TRJ
- VM

VARIABLES

BVAL	--	BIAS IN MEASUREMENT
DUM	--	INTERMEDIATE VARIABLE USED IN COMPUTING EDEVX
I	--	INDEX
ICODE	--	INTERMEDIATE EVENT CODE
IRUN	--	NUMBER OF RUNS MADE
IRUNX	--	TOTAL NUMBER OF RUNS TO BE MADE
J	--	INDEX
K	--	INDEX
NEVENT	--	NUMBER OF EVENTS HAVING OCCURRED
NR	--	NUMBER OF ROWS IN H
TEVN	--	INTERMEDIATE TIME OF EVENT
TRTM2	--	TIME OF NEXT MEASUREMENT

*

```
MAIN--TARGETING
   COMMON
     VMP
     BLK
   VARIABLES
     A        --  SEMI-MAJOR AXIS OF TARGET PLANET HYPERBOLA
     AC       --  ACCURACY LEVELS TO BE USED IN TARGETING
     ACC      --  FINAL ACCURACY LEVEL
     ACK      --  CURRENT ACCURACY LEVEL DURING NUMERICAL
                  DIFFERENCING
     AINCL    --  TRAJECTORY INCLINATION AT CLOSEST APPROACH
     AU       --  CONVERSION UNIT FROM KM TO ASTRONOMICAL UNITS
                  (A.U.)
     AUDAY    --  CONVERSION UNIT FROM KM/SEC TO AU/DAY
     AUS      --  CONVERSION UNIT FROM KM TO A.U.
     BDELV    --  BASIC VELOCITY INCREMENT FOR STM COMPUTATION
     CTSE     --  COSINE OF TRUE ANOMALY AT SPHERE OF INFLUENCE
                  IN TARGET PLANET HYPERBOLA
     DB       --  TARGET IMPACT PARAMETER B
     DBDR     --  B.R
     DBDT     --  B.T
     DELTM    --  DAYS BETWEEN PRINTOUT ON FINAL TRAJECTORY
                  INTEGRATION
     DELV     --  VELOCITY INCREMENT USED IN CURRENT TARGETING
     DEND     --  JULIAN DATE OF FINAL TIME IN INTEGRATION
     DENE     --  INTERMEDIATE VARIABLE
     DINCL    --  TARGET INCLINATION (DEGREES)
     DRCA     --  TARGET RADIUS OF CLOSEST APPROACH
     DT1      --  DESIRED CHANGE IN FIRST TARGET PARAMETER
     DT2      --  DESIRED CHANGE IN SECOND TARGET PARAMETER
     DT3      --  DESIRED CHANGE IN THIRD TARGET PARAMETER
     DV1      --  PREDICTED CHANGE IN FIRST VELOCITY COMPONENT
     DV2      --  PREDICTED CHANGE IN SECOND VELOCITY COMPONENT
     DV3      --  PREDICTED CHANGE IN THIRD VELOCITY COMPONENT
     D1       --  JULIAN DATE OF INITIAL TIME
     D2       --  JULIAN DATE AT SHPERE OF INFLUENCE
     D3       --  JULIAN DATE AT CLOSEST APPROACH
     E        --  ECCENTRICITY OF TARGET PLANET HYPERBOLA
     ECEQP    --  TRANSFORMATION MATRIX FROM TARGET PLANET
                  ECLIPTIC TO TARGET PLANET EQUATORIAL
     EQECP    --  TRANSFORMATION MATRIX FROM TARGET PLANET
                  EQUATORIAL TO TARGET PLANET ECLIPTIC
     ERROR    --  MEASURE OF ERROR IN CURRENT ITERATION
     ERR1     --  ERROR IN FIRST TARGET PARAMETER
     ERR2     --  ERROR IN SECOND TARGET PARAMETER
     ERR3     --  ERROR IN THIRD  TARGET PARAMETER(TARGET DATE)
     FE       --  INTERMEDIATE VARIABLE IN HYPERBOLIC TIME
                  CALCULATION
     HI       --  INCLINATION OF TARGET PLANET HYPERBOLIC
                  W.R.T. ECLIPTIC
     HL       --  LONGITUDE OF TARGET PLANET HYPERBOLIC W.R.T.
                  ECLIPTIC
     HP       --  SEMI-LATUS RECTUM OF TARGET PLANET HYPERBOLA
                  W.R.T. ECLIPTIC
```

*

```
HW       --  ARGUMENT OF PERIAPSIS OF TARGET PLANET
             HYPERBOLA W.R.T. ECLIPTIC
IDAT1    --  CALENDAR DATE OF INITIAL DATE EXCEPT SECONDS
IDAT2    --  CALENDAR DATE AT SPHERE OF INFLUENCE
IDAT3    --  CALENDAR DATE AT CLOSEST APPROACH
ILS6     --  FLAG SPECIFYING WHETHER LAST STEP UNDER
             TARGET OPTION 6 HAS BEEN MADE
INCPR    --  DESIRED NUMBER OF INCREMENTS BETWEEN PRINTOUTS
             OF FINAL TRAJECTORY
INJEK    --  INJECTION OPTION FLAG
IPSI2    --  FLAG INDICATING WHETHER CLOSEST APPROACH
             STM HAS BEEN COMPUTED OR NOT
ISKEJ    --  TOTAL NUMBER OF ACCURACY LEVELS
ISP2     --  FLAG INDICATING WHETHER INTEGRATION SHOULD
             STOP AT SHPERE OF INFLUENCE OR NOT
ISSKJ    --  STORAGE FOR ISKEJ
ISTEP    --  CURRENT STEP (INTEGRATION) DURING STM
             PERTURBED INTEGRATIONS
ISTM     --  FLAG INDICATING WHETHER STM IS RECOMPUTED
             OR NOT
ISTP     --  STORAGE FOR ISTEP
ITARG    --  TARGET OPTION FLAG
ITARGS   --  STORAGE FOR ITARG
ITER     --  NUMBER OF ITERATIONS COMPLETED
ITIM     --  FLAG INDICATING WHETHER TIME OF CLOSEST
             APPROACH HAS BEEN COMPUTED OR  NOT
JC3      --  FLAG INDICATING WHETHER BIASED OR UNBIASED
             PATCHED CONIC CONDITIONS ARE TO BE COMPUTED
JINJT    --  FLAG INDICATING WHETHER INJECTION TIME SHOULD
             BE UPDATED OR NOT
LEVEL    --  CURRENT LEVEL OF ACCURACY IN TARGETING
             PROCEDURE
MIDI     --  NUMBER OF ITERATIONS TO BE MADE AT EACH
             INTERMEDIATE ACCURACY LEVEL
NITRS    --  MAXIMUM NUMBER OF ITERATIONS TO BE MADE AT
             CURRENT LEVEL
NITS     --  MAXIMUM NUMBER OF ITERATIONS TO BE MADE AT
             FINAL LEVEL
NLP      --  INDEX OF LAUNCH PLANET
NOSOI    --  FLAG INDICATING WHETHER IN INNER OR OUTER
             TARGETING
P        --  UNIT VECTOR IN DIRECTION OF PERIAPSIS IN
             TARGET PLANET HYPERBOLA
PERI     --  PERIAPSIS RADIUS OF TARGET PLANET HYPERBOLA
PHI      --  STM FOR TARGETING OPTION 3
PRERR    --  ERROR OF PREVIOUS ITERATE
PSI      --  STM FOR ANY SET OF 3 TARGET CONDITIONS
Q        --  UNIT VECTOR PERPENCICULAR TO P IN TARGET
             PLANET HYPERBOLA
R        --  MAGNITUDE OF RADIUS AT CLOSEST APPROACH
RAD      --  CONVERSION UNIT FROM DEGREES TO RADIUS
RCA      --  TRAJECTORY RADIUS OF CLOSEST APPROACH
RD       --  RATE OF CHANGE OF RADIAL MAGNITUDE AT
             CLOSEST APPROACH BEFORE ITERATING
```

*

RED	--	FACTOR BY WHICH BAD STEP CORRECTIONS ARE REDUCED
RM	--	RADIUS MAGNITUDE
RQ	--	CLOSEST APPRIACH STATE IN TARGET PLANET EQUATORIAL COORDINATES
RS	--	INITIAL STATE IN TRAJECTORY INTEGRATIONS
RSF	--	FINAL STATE IN TRAJECTORY INTEGRATIONS
RSS	--	STORAGE OF TARGETED STATE AT FIRST ACCURACY LEVEL
R1	--	INJECTION POSITION IN LAUNCH PLANET ECLIPTIC COORDIATES
SDEL	--	STORAGE FOR DELV
SFE	--	INTERMEDIATE VARIABLES FOR HYPERBOLIC TIME CALCULATION
SRS	--	STORAGE OF TARGETED VELOCITY AT FINAL ACCURACY LEVEL
SSPH	--	STORAGE FOR TTS
STARG1	--	STORAGE FOR TARG1
STARG2	--	STORAGE FOR TARG2
STARG3	--	STORAGE FOR TARG3
SV1	--	STORAGE OF PREVIOUS ITERATE FIRST VELOCITY COMPONENT
SV2	--	STORAGE OF PREVIOUS ITERATE SECOND VELOCITY COMPONENT
SV3	--	STORAGE OF PREVIOUS ITERATE THIRD VELOCITY COMPONENT
S1	--	SECONDS OF INITIAL TIME
S2	--	SECONDS OF TIME AT SPHERE OF INFLUENCE
S3	--	SECONDS OF TIME AT CLOSEST APPROACH
TA	--	TRUE ANOMALY AT SPHERE OF INFLUENCE ALONG TARGET PLANET HYPERBOLA
TARG1	--	FIRST TARGET VALUE
TARG2	--	SECOND TARGET VALUE
TARG3	--	TARGET TIME
TIMC	--	CUMULATIVE COMPUTER TIME
TIMCR	--	CORRECTION TIME FOR CLOSEST APPROACH ITERATION
TIMPR	--	DAYS BETWEEN PRINTOUTS ON FINAL TRAJECTORY INTEGRATION
TIMS	--	COMPUTER TIME AT START OF PROGRAM
TMDF	--	TIME CORRECTION FOR OUTER TARGETING
TMU	--	TARGET PLANET GRAVITATIONAL CONSTANT
TOLR1	--	TOLERANCE OF FIRST VARIABLE AT CURRENT LEVEL
TOLR2	--	TOLERANCE OF SECOND VARIABLE AT CURRENT LEVEL
TOLR3	--	TOLERANCE OF THIRD VARIABLE AT CURRENT LEVEL
TOL1	--	FINAL TOLERANCE OF FIRST TARGET VARIABLE
TOL2	--	FINAL TOLERANCE OF SECOND TARGET VARIABLE
TOL3	--	FINAL TOLERANCE OF THIRD TARGET VARIABLE
TRG1	--	TRAJECTORY VALUES OF FIRST TARGET PARAMETER AT NOMINAL AND PERTURBED INTEGRATIONS
TRG2	--	TRAJECTORY VALUES OF SECOND TARGET PARAMETER AT NOMINAL AND PERTURBED INTEGRATIONS
TRG3	--	TRAJECTORY VALUES OF THIRD TARGET PARAMETER AT NOMINAL AND PERTURBED INTEGRATIONS

*

TRTM	--	INITIAL TRAJECTORY TIME
TSICA	--	TIME FROM SPHERE OF INFLUENCE TO CLOSEST APPROACH
TSPH	--	TARGET PLANET SPHERE OF INFLUENCE
V	--	VELOCITY MAGNITUDE AT CLOSEST APPROACH IN OUTER TARGETING
VEL	--	VELOCITY COMPONENTS FOR NOMINAL AND PERTURBED INTEGRATIONS
VHE	--	HYPERBOLIC EXCESS VELOCITY ON TARGET PLANET HYPERBOLA
VM	--	VELOCITY MAGNITUDE AT CLOSEST APPROACH
VX	--	MAGNITUDE OF VELOCITY AT SPHERE OF INFLUENCE ON TARGET PLANET HYPERBOLA
V1	--	INJECTION VELOCITY
WPT	--	NORMAL TO TARGET PLANET HYPERBOLA
XTOL1	--	STORAGE FOR TOL1
XTOL2	--	STORAGE FOR TOL2
XTOL3	--	STORAGE FOR TOL3
Z	--	NORMAL TO CLOSEST APPROACH PLANE
ZM	--	MAGNITUDE OF ANGULAR MOMENTUM AT CLOSEST APPROACH

1. ACTB

VARIABLES

A	--	SEMI-MAJOR AXIS
AB	--	MAGNITUDE OF RV
BV	--	B VECTOR
CTA	--	COSINE OF TRUE ANOMALY
C1	--	MAGNITUDE OF RXV
E	--	ECCENTRICITY
I	--	INDEX
P	--	SEMI-LATUS RECTUM
PV	--	INTERMEDIATE VECTOR
QV	--	INTERMEDIATE VECTOR
RD	--	R DOT V DIVIDED BY MAGNITUDE OF R
RM	--	MAGNITUDE OF R
RRD	--	R DOT V
RV	--	INTERMEDIATE VECTOR
STA	--	SINE OF TRUE ANOMALY
SV	--	INTERMEDIATE VECTOR
TV	--	INTERMEDIATE VECTOR
VM	--	MAGNITUDE OF V
WV	--	RXV
Z	--	INTERMEDIATE VECTOR

2. AUX

VARIABLES

ANG1	--	ANGLE OF FIRST BURN
ANG2	--	ANGLE OF SECOND BURN
AZ	--	DESIRED LAUNCH AZIMUTH
AZI	--	INJECTION AZIMUTH
DJL	--	JULIAN DATE OF LAUNCH
E	--	ECCENTRICITY OF HYPERBOLA
ELAT	--	LATITUDE OF LAUNCH SITE

*

ELON -- LONGITUDE OF LAUNCH SITE
GME -- GRAVITATIONAL CONSTANT OF LAUNCH PLANET
PHI -- INJECTION LATITUDE
PV -- UNIT VECTOR IN DIRECTION OF PERIAPSIS OF HYPERBOLA
Q -- UNIT VECTOR IN PLANE OF HYPERBOLA PERPENDICULAR TO PV
RAI -- INJECTION RIGHT ASCENSION
ROT -- ROTATIONAL RATE OF LAUNCH PLANET
RP -- PERIAPSIS RADIUS OF HYPERBOLA
S -- UNIT VECTOR IN DIRECTION OF DEPARTURE ASYMPTOTE
TAI -- TRUE ANOMALY AT INJECTION
TB -- TIME BETWEEN LAUNCH AND INJECTION
TC -- COAST TIME
THI -- INJECTION LONGITUDE
TIM1 -- TIME OF FIRST BURN (SEC)
TIM2 -- TIME OF SECOND BURN (SEC)
TINJ -- TIME (HRS) OF INJECTION FROM ZERO HOURS ON DATE OF LAUNCH
TL -- TIME (HRS) OF LAUNCH AFTER ZERO HOURS ON DATE OF LAUNCH
W -- NORMAL TO LAUNCH PLANE

3. BIAS
COMMON
MISC
VARIABLES
BVAL -- BIASES
MCODE -- MEASUREMENT CODE

4. BLOCK DATA
COMMON
BLK
COM
PRT

5. CASOI
VARIABLES
A -- SEMI-MAJOR AXIS
C -- NORMAL TO PLANE IN TARGET PLANE EQUATORIAL COORDINATES
CIMP -- NORMAL TO PLANE IN IMPACT PLANE COORDINATES (R HAT, S HAT, T HAT)
CM -- ANGULAR MOMENTUM FROM EQUATORIAL COORDINATES
CSD -- COSINE (DVP)
CSW -- COSINE (RVP)
CTA -- COSINE (TA)
CTS -- COSINE OF TRUE ANOMALY AT SPHERE OF INFLUENCE
C1 -- ANGULAR MOMENTUM FROM ECLIPTIC COORDINATES
DB -- TARGET IMPACT PARAMETER B
DBDR -- TARGET D.R
DBDT -- TARGET B.T
DECL -- DECLINATION OF APPROACH ASYMPTOTE (EQUATORIAL COORDINATES)

*

DEN -- INTERMEDIATE VARIABLE IN HYPERBOLIC TIME CALCULATION
DINCL -- INCLINATION AT CLOSEST APPROACH
DRCA -- RADIUS AT CLOSEST APPROACH
DVP -- DECLINATION OF APPROACH ASYMPTOTE (ECLIPTIC COORDINATES)
E -- ECCENTRICITY OF TARGET PLANE HYPERBOLA
ECIM -- ECLIPTIC-TO-IMPACT PLANE COORDINATE TRANSFORMATION
EQEC -- TRANSFORMATION MATRIX FROM EQUATORIAL TO ECLIPTIC COORDINATES
EQIM -- EQUATORIAL-TO-IMPACT PLANE COORDINATE TRANSFORMATION
F -- INTERMEDIATE VARIABLE IN HYPERBOLIC TIME CALCULATION
HI -- INCLINATION OF HYPERBOLIC ORBIT PLANE
HL -- LONGITUDE OF ASCENDING NODE OF HYPERBOLIC ORBIT PLANE
HP -- SEMI-LATUS RECTUM OF HYPERBOLA
HW -- ARGUMENT OF PERIAPSIS OF HYPERBOLA
P -- UNIT VECTOR IN DIRECTION OF PERIAPSIS
PERI -- PERIAPSIS RADIUS OF HYPERBOLA
PI -- 3.1415926536
Q -- UNIT VECTOR NORMAL TO P BAR IN ORBITAL PLANE
RA -- LONGITUDE OF INCOMING ASYMPTOTE (EQUATORIAL COORDINATES)
RAD -- CONVERSION UNIT FROM RADIANS TO DEGREES
RD -- INTERMEDIATE VARIABLE IN HYPERBOLIC TIME CALCULATION
RRD -- INTERMEDIATE VARIABLE IN HYPERBOLIC TIME CALCULATION
RS -- POSITION VECTOR AT SPHERE OF INFLUENCE
RSM -- RADIUS MAGNITUDE AT SPHERE OF INFLUENCE INTERSECTION
RVP -- RIGHT ASCENSION OF INCOMING ASYMPTOTIC (ECLIPTIC COORDINATES)
S -- INCOMING ASYMPTOTE(EQUATORIAL COORDINATES)
SCZ -- SIGN OF Z COMPONENT OF ANGULAR MOMENTUM
SD -- SIGN OF DECLINATION DECL
SDELW -- SIGN OF ANGLE BETWEEN ASCENDING NODE AND INCOMING ASYMPTOTE
SF -- INTERMEDIATE VARIABLE IN HYPERBOLIC TIME CALCULATION
SI -- SIGN OF TARGET INCLINATION
SINCL -- STORAGE OF DINCL
SM -- SIGN OF MOTION (POSIGRADE OR RETROGRADE)
SND -- SINE(DVP)
SNW -- SINE RVP
STA -- SINE TA
STS -- SINE OF ANGLE AT SPHERE OF INFLUENCE
TA -- TRUE ANOMALY AT SPHERE OF INFLUENCE
TAND -- TANGENT DVP
TDELW -- TANGENT OF ANGLE BETWEEN ASCENDING NODE AND INCOMING ASYMPTOTE

*

```
THETA    --  ANGLE IN B-PLANE FROM T-AXIS TO B-PARAMETER
TSICA    --  TIME FROM SPHERE OF INFLUENCE TO CLOSEST
             APPROACH ON HYPERBOLA
TTG      --  GRAVITATIONAL CONSTANT OF TARGET PLANET
VHE      --  CORRECTED HYPERBOLIC EXCESS VELOCITY
VHP      --  VELOCITY VECTOR AT SPHERE OF INFLUENCE
VHPM     --  MAGNITUDE OF APPROACH ASYMPTOTE
VVP      --  NORMALIZED APPROACH ASYMPTOTE (ECLIPTIC COOR)
W        --  LONGITUDE OF ASCENDING NODE (ECLIPTIC
             COORDINATES)
WPT      --  NORMAL TO ORBITAL PLANE (ECLIPTIC COORDINATE)
Z        --  INTERMEDIATE VECTOR USED IN COMPUTING P,Q
             VECTORS
```

6. CONC2

```
VARIABLES
A        --  SEMI-MAJOR AXIS
AM2      --  INTERMEDIATE VARIABLE
A1       --  INTERMEDIATE VARIABLE
A2       --  INTERMEDIATE VARIABLE
A3       --  INTERMEDIATE VARIABLE
CSE      --  COSINE OF ECCENTRIC ANOMALY
CTA      --  COSINE OF TRUE ANOMALY
CTA2     --  COSINE OF TRUE ANOMALY ON ELLIPSE
C1       --  MAGNITUDE OF RXV
DDXO     --  INTERMEDIATE VARIABLE
DDYO     --  INTERMEDIATE VARIABLE
DXO      --  INTERMEDIATE VARIABLE
DYO      --  INTERMEDIATE VARIABLE
E        --  ECCENTRICITY
EA       --  ECCENTRIC ANOMALY
F        --  INTERMEDIATE VARIABLE
FMI1     --  INTERMEDIATE VECTOR
FMI      --  INTERMEDIATE VECTOR
I        --  INDEX
J        --  INDEX
K        --  INDEX
L        --  INDEX
N        --  INTERMEDIATE VARIABLE
OPEC     --  INTERMEDIATE VECTOR
ORB      --  INTERMEDIATE VARIABLE
P        --  SEMI-LATUS RECTUM
PI       --  MATHEMATICAL CONSTANT
PSIOP    --  INTERMEDIATE STATE TRANSITION MATRIX
PV       --  INTERMEDIATE VECTOR
Q        --  INTERMEDIATE VECTOR
RD       --  R DOT V DIVIDED BY MAGNITUDE OF R
RM       --  MAGNITUDE OF R
RRD      --  R DOT V
RTHD     --  INTERMEDIATE VARIABLE
R2       --  INTERMEDIATE VARIABLE
R3       --  INTERMEDIATE VARIABLE
SNE      --  SINE OF ECCENTRIC ANOMALY
SNF      --  SINE OF F
```

*

```
    STA      --  SINE OF TRUE ANOMALY
    STA2     --  SINE OF TRUE ANOMALY ON ELLIPSE
    TIM1     --  INTERMEDIATE TIME
    TIM2     --  INTERMEDIATE TIME
    VM       --  MAGNITUDE OF V
    WV       --  RXV
    XO       --  INTERMEDIATE VARIABLE
    YO       --  INTERMEDIATE VARIABLE
    Z        --  INTERMEDIATE VECTOR

7.  CONIC
    VARIABLES
    A        --  SEMI-MAJOR AXIS
    E        --  ECCENTRICITY
    GMX      --  GRAVITATIONAL CONSTANT OF PRIMARY BODY
    P        --  SEMI-LATUS RECTUM
    PV       --  STANDARD UNIT VECTOR IN DIRECTION OF
                 PERIAPSIS
    Q        --  STANDARD UNIT VECTOR IN ORBITAL PLANE
                 NORMAL TO PV
    R        --  POSITION VECTOR
    RM       --  POSITION VECTOR MAGNITUDE
    RP       --  PERIAPSIS RADIUS
    TA       --  TRUE ANOMALY OF SPECIFIED POSITION
    V        --  VELOCITY VECTOR
    VM       --  VELOCITY VECTOR MAGNITUDE
    W        --  ARGUMENT OF PERIAPSIS
    WV       --  NORMAL TO ORBITAL PLANE
    XI       --  INCLINATION
    XL       --  LONGITUDE OF ASCENDING NODE

8.  CONST
    VARIABLES
    ANG1     --  ANGLE OF FIRST BURN
    ANG2     --  ANGLE OF SECOND BURN
    DDIQ     --  OBLIQUITY OF LAUNCH PLANET ORBIT
    DDLAT    --  LATITUDE OF LAUNCH SITE
    DDLON    --  LONGTUDE OF LAUNCH SITE
    DDLQ     --  ASCENDING NODE OF LAUNCH PLANET ORBIT
    HHTA     --  TRUE ANOMALY OF TRAJECTORY
    NDD      --  INDEX OF LAUNCH PLANET
    NTT      --  INDEX OF TARGET PLANET
    ROT      --  ROTATIONAL RATE OF LAUNCH PLANET
    RP       --  PARKING ORBIT RADIUS
    TIM1     --  TIME OF FIRST BURN
    TIM2     --  TIME OF SECOND BURN

9.  CONVERT
    VARIABLES
    B1       --  INTERMEDIATE VARIABLE
    B2       --  INTERMEDIATE VARIABLE
    B3       --  INTERMEDIATE VARIABLE
    CG       --  COSINE OF PATH ANGLE
    CP       --  COSINE OF DECLINATION
```

*

```
      CT       --  COSINE OF RIGHT ASCENSION
      SG       --  SINE OF PATH ANGLE
      SP       --  SINE OF DECLINATION
      ST       --  SINE OF RIGHT ASCENSION

10. DATA
    COMMON
      BLK
      COM
      CONST
      CONST2
      CONST3
      EVENT
      GUI
      MEAS
      MISC
      NAME
      PRT
      SIMCNT
      SIM1
      SIM2
      STM
      STVEC
      TIM
      TRAJCD
      TRJ
      VM
    VARIABLES
      AMIN     --  INTERMEDIATE VARIABLE
      D        --  INTERMEDIATE JULIAN DATE
      DATE     --  INTERMEDIATE JULIAN DATE
      EARTH    --  CALENDAR DATE AT WHICH EARTHS ORBITAL ELEMENTS
                   WILL BE CALCULATED
      FNDT     --  DATE OF FINAL TIME
      GAMMA    --  PATH ANGLE
      I        --  INDEX
      ICNT     --  COUNTER
      IDAY     --  CALENDAR DAY OF FINAL TIME
      IHR      --  CALENDAR HOUR OF FINAL TIME
      IL       --  INDEX
      IMIN     --  CALENDAR MINUTES OF FINAL TIME
      IMO      --  CALENDAR MONTH OF FINAL TIME
      IPRO     --  PROBLEM NUMBER
      IROW     --  INDEX
      IYR      --  CALENDAR YEAR OF FINAL TIME
      I1       --  COUNTER
      I2       --  COUNTER
      I3       --  COUNTER
      I4       --  COUNTER
      J        --  INDEX
      JUPITER  --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
                   JUPITER WILL BE CALCULATED
      K        --  INDEX
      LDAY     --  CALENDAR DAY OF INITIAL TIME
```

*

```
LHR      --  CALENDAR HOURS OF INITIAL TIME
LMIN     --  CALENDAR MINUTES OF INITIAL TIME
LMO      --  CALENDAR MONTH OF INITIAL TIME
LYR      --  CALENDAR YEAR OF INITIAL TIME
MARS     --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             MARS WILL BE CALCULATED
MEAS     --  MEASUREMENT CODES
MERCURY  --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             MERCURY WILL BE CALCULATED
MOON     --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             EARTHS MOON WILL BE CALCULATED
NENT     --  NUMBER OF ENTRIES IN MEASUREMENT SCHEDULE
NEPTUNE  --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             NEPTUNE WILL BE CALCULATED
N1       --  COUNTER
N2       --  COUNTER
N3       --  COUNTER
N4       --  COUNTER
PHI      --  DECLINATION
PLUTO    --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             PLUTO WILL BE CALUCLATED
RDS      --  GEOCENTRIC RADIUS OF VEHICLE
SATURN   --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             SATURN WILL BE CALCULATED
SCHED    --  ARRAY OF TIMES IN MEASUREMENT SCHEDULE
SEC      --  INTERMEDIATE CALENDAR SECONDS
SECI     --  CALENDAR SECONDS AT FINAL TIME
SECL     --  CALENDAR SECONDS AT INITIAL TIME
SIGMA    --  AZIMUTH
THETA    --  RIGHT ASCENSION
T1       --  INTERMEDIATE TIME
T2       --  INTERMEDIATE TIME
T3       --  INTERMEDIATE TIME
T4       --  INTERMEDIATE TIME
URANUS   --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             URANUS WILL BE CALCULATED
VEL      --  MAGNITUDE OF VELOCITY OF VEHICLE
VENUS    --  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF
             VENUS WILL BE CALCULATED
VUNIT    --  INTERMEDIATE VELOCITY CONVERSION FACTOR
```

```
11. DYNO
    COMMON
     CONST
     MISC
     SIMCNT
     SIM1
     STM
     STVEC
     TIM
     VM
    VARIABLES
     DT       --  TIME INCREMENT
     D2       --  SQUARE OF DT
```

*

```
     I        --   INDEX
     IC       --   INDEX
     J        --   INDEX
     T1       --   INITIAL TIME
     T2       --   FINAL TIME

12. EIGEN
    COMMON
     CONST2
     EVENT
     MISC
     NAME
     STM
     STVEC
     TIM
     TRAJCD
     VM
    VARIABLES
     EGVCT    --   EIGENVECTORS
     EGVL     --   EIGENVALUES
     I        --   INDEX
     ICODE    --   INTERMEDIATE CODE USED IN COMPUTING EIGEN-
                   VECTORS
     J        --   INDEX
     K        --   INDEX
     LINES    --   LINE COUNT
     MAX      --   MAXIMUM NUMBER OF LINES PER PAGE
     PEIG     --   INTERMEDIATE ARRAY
     RF       --   POSITION AND VELOCITY OF VEHICLE AT TIME OF
                   EIGENVECTOR EVENT
     RHO      --   CORRELATION COEFFICIENT MATRIX
     VEIG     --   INTERMEDIATE VECTOR

13. EIGSIM
    COMMON
     CONST
     CONST2
     EVENT
     MISC
     NAME
     SIM1
     SIM2
     STVEC
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     DUM      --   INTERMEDIATE VECTOR
     EGVCT    --   EIGENVECTORS
     EGVL     --   EIGENVALUES
     I        --   INDEX
     ICODE    --   INTERMEDIATE CODE USED IN COMPUTING EIGEN-
                   VALUES AND EIGENVECTORS
     J        --   INDEX
     K        --   INDEX
```

*

```
LINES   --  LINE COUNT
MAX     --  MAXIMUM NUMBER OF LINES PER PAGE
PEIG    --  INTERMEDIATE ARRAY
RF      --  POSITION AND VELOCITY OF VEHICLE AT TIME OF
            EIGENVECTOR EVENT ON ORIGINAL NOMINAL
RF1     --  POSITION AND VELOCITY OF VEHICLE ON MOST
            RECENT NOMINAL AT TIME OF EIGENVECTOR EVENT
RF2     --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
            TRAJECTORY AT TIME OF EIGENVECTOR EVENT
RHO     --  CORRELATION COEFFICIENT MATRIX
RI2     --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
            TRAJECTORY AT TIME OF LAST MEASUREMENT OR
            EVENT
VEIG    --  INTERMEDIATE VECTOR
```

```
14. EPHEM
    COMMON
     BLK
     COM
     PRT
    VARIABLES
     DD      --  INTERMEDIATE VARIABLE
     E2      --  INTERMEDIATE VARIABLE
     E3      --  INTERMEDIATE VARIABLE
     FCTR    --  VELOCITY DIVIDED BY RADIUS
     I       --  CODE OF PLANET
     IJ      --  INDEX
     IJ1     --  INDEX
     IK      --  INDEX
     IL      --  INDEX
     IN      --  INDEX, ROW OF F
     IND     --  INDEX
     ITEMP   --  INTERMEDIATE VARIABLE
     ITEST   --  INTERNAL CODE WHICH DETERMINES IF COORDINATES
                 OF EARTH ARE BEING CALCULATED ONLY IN ORDER
                 TO COMPUTE THOSE OF MOON
     ITEST2  --  INTERNAL CODE WHICH DETERMINES IF COORDINATES
                 OF EARTH HAVE BEEN COMPUTED PRIOR TO COMPUTING
                 THOSE OF MOON
     J       --  INDEX
     K       --  INDEX
     P       --  SEMI-LATUS RECTUM
     PI2     --  EQUAL TO TWO TIMES PI
     R       --  HELIOCENTRIC RADIUS OF PLANET
     TRG     --  ARRAY OF TRIGONOMETRIC FUNCTIONS OF SPECIFIED
                 ANGLES
     VEL     --  VELOCITY OF PLANET
     WX      --  X-COMPONENT OF INTERMEDIATE VECTOR, W
     WY      --  Y-COMPONENT OF INTERMEDIATE VECTOR, W
     WZ      --  Z-COMPONENT OF INTERMEDIATE VECTOR, W
```

```
15. ESTMT
    COMMON
     BLK
```

*

```
     COM
     PRT
    VARIABLES
     I        --   INDEX
     J        --   INDEX

16.  EULMX
    VARIABLES
     ALP      --   FIRST ROTATION ANGLE
     BET      --   SECOND ROTATION ANGLE
     GAM      --   THIRD ROTATION ANGLE
     LL       --   THIRD AXIS OF ROTATION
     MM       --   SECOND AXIS OF ROTATION
     NN       --   FIRST AXIS OF ROTATION
     P        --   THE TRANSFORMATION MATRIX

17.  GHA
    COMMON
     TIM
    VARIABLES
     D        --   NUMBER OF DAYS IN TSTAR
     EQMEG    --   EARTH ROTATION RATE
     GH       --   GREENWICH HOUR ANGLE
     ID       --   INTERMEDIATE VARIABLE
     REFJD    --   JULIAN DATE OF JAN. 0, 1950
     TFRAC    --   FRACTION OF DAY IN TSTAR
     TSTAR    --   JULIAN DATE, EPOCH JAN. 0, 1950, OF INITIAL
                   TRAJECTORY TIME

18.  GUID
    COMMON
     BLK
     CONST2
     EVENT
     GUI
     MISC
     NAME
     STM
     STVEC
     TIM
     TRAJCD
     VM
    VARIABLES
     A        --   GUIDANCE SUB-MATRIX A
     BB       --   GUIDANCE SUB-MATRIX B
     BDRS     --   B DOT R
     BDR1     --   VALUE OF B DOT R RETURNED FROM PARTL (NOT
                   USED)
     BDTS     --   B DOT T
     BDT1     --   VALUE OF B DOT T RETURNED FROM PARTL (NOT
                   USED)
     BS       --   MAGNITUDE OF B VECTOR
     B1       --   VALUE OF B RETURNED FROM PARTL (NOT USED)
     D        --   INTERMEDIATE JULIAN DATE
```
*

DUM1	--	EIGENVECTORS
D1	--	STORAGE LOCATION FOR DTMAX
EGVCT	--	EIGENVECTORS
EGVL	--	EIGENVALUES
I	--	INDEX
ICLS	--	INTERMEDIATE STORAGE FOR ICL
ICS	--	INTERMEDIATE STORAGE FOR ICL2
INCMTS	--	INTERMEDIATE STORAGE FOR INCMT
IPR	--	STORAGE FOR IPRINT
ISP	--	INTERMEDIATE STORAGE FOR ISP2
J	--	INDEX
K	--	INDEX
L	--	INDEX
LINES	--	LINE COUNT
MAX	--	MAXIMUM NUMBER OF LINES PER PAGE
PBR	--	PARTIAL OF B DOT R WITH RESPECT TO STATE VECTOR
PBT	--	PARTIAL OF B DOT T WITH RESPECT TO STATE VECTOR
PHI1	--	INTERMEDIATE ARRAY
PHI2	--	INTERMEDIATE ARRAY
PHI3	--	INTERMEDIATE ARRAY
RI	--	POSITION AND VELOCITY OF VEHICLE AT TIME OF GUIDANCE EVENT
RTPS	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO SUN AT SPHERE OF INFLUENCE
TCA	--	TRAJECTORY TIME AT CLOSEST APPROACH
TIMINS	--	INTERMEDIATE STORAGE FOR TIMINT
TSI	--	TRAJECTORY TIME AT SPHERE OF INFLUENCE
XCA	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO SUN AT CLOSEST APPROACH
XSIP	--	POSITION OF VEHICLE RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE
XSIV	--	VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE

19. GUIDM

COMMON

CONST
CONST2
EVENT
GUI
MISC
NAME
STM
STVEC
TIM
TRAJCD
VM

VARIABLES

ADA	--	VARIATION MATRIX
AMAX	--	INTERMEDIATE VARIABLE USED TO FIND MAXIMUM EIGENVALUE OF S
DUM	--	RHO DIVIDED BY MAGNITUDE OF EIGENVECTOR CORRESPONDING TO MAXIMUM EIGENVALUE OF S

*

```
DUM1     --  INTERMEDIATE VARIABLE
DUM2     --  INTERMEDIATE VARIABLE
EGM      --  MAGNITUDE OF EIGENVECTOR CORRESPONDING TO
             MAXIMUM EIGENVALUE OF S
EGVCT    --  EIGENVECTORS
EGVL     --  EIGENVALUES
EXEC     --  EXECUTION ERROR MATRIX
EXM      --  INTERMEDIATE VARIABLE
EXV      --  EXPECTED VALUE OF VELOCITY CORRECTION
GA       --  GUIDANCE MATRIX
GAP      --  INTERMEDIATE ARRAY EQUAL TO GA TIMES P
I        --  INDEX
ICODE    --  INTERNAL CODE
ICODE2   --  INTERMEDIATE CODE
IGP      --  CODE WHICH DETERMINES WHICH GUIDANCE POLICY
             IS BEING USED
IQP      --  CODE WHICH DETERMINES WHICH POLICY IS BEING
             USED TO COMPUTE THE EXECUTION ERROR MATRIX
ISPHC    --  TEMPORARY STORAGE FOR ISPH
ITEMP    --  INTERMEDIATE VARIABLE
J        --  INDEX
K        --  INDEX
L        --  INDEX
LINES    --  LINE COUNT
MAP      --  INDEX OF MAXIMUM EIGENVALUE OF S
MAX      --  MAXIMUM LINES PER PAGE
NN       --  INTERMEDIATE INDEX
P1       --  INTERMEDIATE STORAGE FOR P
P2       --  INTERMEDIATE STORAGE FOR P
RF       --  POSITION AND VELOCITY OF VEHICLE AT TIME OF
             GUIDANCE EVENT
RHO      --  EXPECTED VALUE OF DELTA V
S        --  COVARIANCE MATRIX ASSOCIATED WITH VELOCITY
             COMPONENTS
SDV      --  STANDARD DEVIATION OF EXPECTED VALUE OF
             DELTA V
TRS      --  TRACE OF S
U        --  INTERMEDIATE VARIABLE
VEIG     --  INTERMEDIATE VECTOR
X2       --  INTERMEDIATE VARIABLE
Y2       --  INTERMEDIATE VARIABLE
Z2       --  INTERMEDIATE VARIABLE
```

```
20. GUIS
    COMMON
     BLK
     CONST
     CONST2
     EVENT
     GUI
     MISC
     NAME
     SIM1
     SIM2
```

*

```
 STM
 STVEC
 TIM
 TRAJCD
 TRJ
 VM
VARIABLES
 A        --  GUIDANCE SUB-MATRIX A
 BB       --  GUIDANCE SUB-MATRIX B
 BDR1     --  VALUE OF B DOT R RETURNED FROM PARTL (NOT
              USED)
 BDT1     --  VALUE OF B DOT T RETURNED FROM PARTL (NOT
              USED)
 B1       --  MAGNITUDE OF B VECTOR RETURNED FROM PARTL
              (NOT USED)
 DUM      --  INTERMEDIATE ARRAY
 DUM1     --  EIGENVECTORS
 D1       --  TEMPORARY STORAGE FOR DTMAX
 EGVCT    --  EIGENVECTORS
 EGVL     --  EIGENVALUES
 I        --  INDEX
 ICLS     --  INTERMEDIATE STORAGE FOR ICL
 ICS      --  INTERMEDIATE STORAGE FOR ICL2
 IPR      --  INTERMEDIATE STORAGE FOR IPRINT
 ISPS     --  INTERMEDIATE STORAGE FOR ISP2
 J        --  INDEX
 K        --  INDEX
 L        --  INDEX
 LINES    --  LINE COUNT
 MAX      --  MAXIMUM NUMBER OF LINES PER PAGE
 PBR      --  PARTIAL OF B DOT R WITH RESPECT TO STATE
              VECTOR
 PBT      --  PARTIAL OF B DOT T WITH RESPECT TO STATE
              VECTOR
 PHI1     --  INTERMEDIATE ARRAY
 PHI2     --  INTERMEDIATE ARRAY
 PHI3     --  INTERMEDIATE ARRAY
 RI       --  POSITION AND VELOCITY OF VEHICLE ON ORIGINAL
              NOMINAL TRAJECTORY AT TIME OF GUIDANCE EVENT
 RI1      --  POSITION AND VELOCITY OF VEHICLE ON MOST
              RECENT NOMINAL AT TIME OF GUIDANCE EVENT
 RMCA     --  DISTANCE OF VEHICLE FROM TARGET PLANET AT
              CLOSEST APPROACH
 RMSI     --  DISTANCE OF VEHICLE FROM TARGET PLANET AT
              SPHERE OF INFLUENCE
 RTPS     --  POSITION AND VELOCITY OF VEHICLE RELATIVE
              TO SUN AT SPHERE OF INFLUENCE
 TCA      --  TRAJECTORY TIME AT CLOSEST APPROACH
 TSI      --  TRAJECTORY TIME AT SPHERE OF INFLUENCE
 VMCA     --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE
              TO TARGET PLANET AT CLOSEST APPROACH
 VMSI     --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE
              TO TARGET PLANET AT SPHERE OF INFLUENCE
 XCA      --  POSITION AND VELOCITY OF VEHICLE RELATIVE
              TO SUN AT CLOSEST APPROACH
```

*

21. GUISIM

COMMON

CONST2
EVENT
GUI
MISC
NAME
SIMCNT
SIM2
STM
STVEC
TIM
TRAJCD
TRJ
VM

VARIABLES

ADA	--	VARIATION MATRIX
AK1	--	INTERMEDIATE VARIABLE FOR RESOLUTION ERROR
AL1	--	INTERMEDIATE VARIABLE FOR ERROR IN POINTING ANGLE 1
BT1	--	INTERMEDIATE VARIABLE FOR ERROR IN POINTING ANGLE 2
DELX	--	ESTIMATED DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL, ALSO USED FOR ERROR AT TARGET CONDITIONS DUE TO NAVIGATION UNCERTAINTY
DUM	--	INTERMEDIATE VECTOR
DUM1	--	INTERMEDIATE VARIABLE
DUM2	--	INTERMEDIATE ARRAY
DUM3	--	INTERMEDIATE VARIABLE
DV	--	PERFECT CORRECTION, ALSO USED FOR ACTUAL ERROR IN CORRECTION AND ACTUAL ERROR AT TARGET AFTER CORRECTION
DVC	--	COMMANDED CORRECTION
DVCM	--	MAGNITUDE OF COMMANDED CORRECTION
DVE	--	ERROR IN CORRECTION DUE TO NAVIGATION UNCERTAINTY, ALSO USED FOR ACTUAL CORRECTION
DX	--	ACTUAL DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL, ALSO USED AS ACTUAL ORBIT DETERMINATION INACCURACY
EGVCT	--	EIGENVECTORS
EGVL	--	EIGENVALUES
EXEC	--	EXECUTION ERROR MATRIX
EXM	--	INTERMEDIATE VARIABLE
GA	--	GUIDANCE MATRIX
GAP	--	INTERMEDIATE ARRAY EQUAL TO GA TIMES P
I	--	INDEX
ICODE	--	INTERNAL CODE
ICODE2	--	INTERNAL CODE
IGP	--	CODE USED TO DETERMINE WHICH GUIDANCE POLICY IS BEING USED
J	--	INDEX
K	--	INDEX
L	--	INDEX

*

```
LINES    --  LINE COUNT
MAX      --  MAXIMUM NUMBER OF LINES PER PAGE
NN       --  INTERMEDIATE INDEX
P1       --  TEMPORARY STORAGE FOR P
RF       --  POSITION AND VELOCITY OF VEHICLE ON ORIGINAL
             NOMINAL AT TIME OF GUIDANCE EVENT
RF1      --  POSITION AND VELOCITY OF VEHICLE ON MOST
             RECENT NOMINAL AT TIME OF GUIDANCE EVENT
RF2      --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
             TRAJECTORY AT TIME OF GUIDANCE EVENT
RI2      --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
             TRAJECTORY AT TIME OF LAST MEASUREMENT OR
             EVENT
S        --  COVARIANCE MATRIX ASSOCIATED WITH VELOCITY
             COMPONENTS
S1       --  INTERMEDIATE VARIABLE FOR PROPORTIONALITY
             ERROR
VEIG     --  INTERMEDIATE VECTOR
X2       --  INTERMEDIATE VARIABLE
Y2       --  INTERMEDIATE VARIABLE
Z2       --  INTERMEDIATE VARIABLE

22. HYELS
    VARIABLES
     I       --  INDEX
     J       --  INDEX
     K       --  INDEX
     K2      --  SQUARE OF SIGMA LEVEL
     PI      --  INVERSE OF MATRIX P
     P12     --  TWICE THE VALUE OF (1,2) ELEMENT OF PI
     P13     --  TWICE THE VALUE OF (1,3) ELEMENT OF PI
     P23     --  TWICE THE VALUE OF (2,3) ELEMENT OF PI
     V       --  TEMPORARY STORAGE VECTOR FOR ARRAY P

23. HYPER
    VARIABLES
     A       --  SEMI-MAJOR AXIS
     AZ      --  LAUNCH AZIMUTH
     C3      --  LAUNCH ENERGY
     DLA     --  DECLINATION OF DEPARTURE ASYMPTOTE
     E       --  ECCENTRICITY
     ELAT    --  LATITUDE OF LAUNCH SITE
     GME     --  GRAVITATIONAL CONSTANT OF LAUNCH PLANET
     P       --  SEMI-LATUS RECTUM
     PV      --  UNIT VECTOR DIRECTED TOWARD PERIAPSIS
     Q       --  UNIT VECTOR NORMAL TO P HAT IN ORBITAL PLANE
     RAL     --  RIGHT ASCENSION OF DEPARTURE ASYMPTOTE
     RP      --  PERIAPSIS RADIUS
     S       --  HYPERBOLIC EXCESS VELOCITY
     VHL     --  SPEED AT INFINITY ALONG HYPERBOLA
     W       --  NORMAL TO PLANE
     XI      --  INCLINATION
     XL      --  LONGITUDE OF ASCENDING NODE
     XW      --  ARGUMENT OF PERIAPSIS
```

*

24. HYPSV

VARIABLES

```
C3      --  ENERGY
E       --  ECCENTRICITY
ECEQ    --  TRANSFORMATION MATRIX FROM ECLIPTIC TO
            EQUATORIAL COORDINATES
GAM     --  PATH ANGLE AT GIVEN RADIUS
GME     --  GRAVITATIONAL CONSTANT OF PRIMARY
P       --  SEMI-LATUS RECTUM
PV      --  UNIT VECTOR TO PERIAPSIS FROM PRIMARY
Q       --  UNIT VECTOR NORMAL TO P HAT IN PLANE OF ORBIT
R       --  RADIUS AT WHICH STATE IS DESIRED
RP      --  PERIAPSIS RADIUS
TA      --  TRUE ANOMALY AT GIVEN RADIUS
TS      --  TIME FROM PERIAPSIS TO RADIUS
VEC     --  VELOCITY AT GIVEN RADIUS
VEQ     --  VELOCITY VECTOR AT GIVEN RADIUS
VHL     --  HYPERBOLIC EXCESS VELOCITY
VS      --  SPEED AT GIVEN RADIUS
XEC     --  POSITION VECTOR AT GIVEN RADIUS
XEQ     --  POSITION VECTOR AT GIVEN RADIUS
```

25. INPUTZ

COMMON

```
BLK
COM
PRT
```

VARIABLES

```
D       --  JULIAN DATE OF INITIAL TRAJECTORY TIME
D2      --  JULIAN DATE OF FINAL TRAJECTORY TIME
I       --  INDEX
IDAY    --  DAY OF CALENDAR DATE OF FINAL TRAJECTORY TIME
IHR     --  HOUR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
IMIN    --  MINUTE OF CALENDAR DATE OF FINAL TRAJECTORY
            TIME
IMO     --  MONTH OF CALENDAR DATE OF FINAL TRAJECTORY
            TIME
INERR   --  INTERNAL CODE
IP      --  CODE NUMBER OF PLANET
IYR     --  YEAR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
J       --  INDEX
LDAY    --  DAY OF CALENDAR DATE OF INITIAL TIME
LHR     --  HOUR OF CALENDAR DATE OF INITIAL TIME
LMIN    --  MINUTE OF CALENDAR DATE OF INITIAL TIME
LMO     --  MONTH OF CALENDAR DATE OF INITIAL TIME
LYR     --  YEAR OF CALENDAR DATE OF INITIAL TIME
SECI    --  SECOND OF CALENDAR DATE OF FINAL TIME
SECL    --  SECOND OF CALENDAR DATE OF INITIAL TIME
TP      --  INTERMEDIATE VARIABLE
```

26. JACOBI

VARIABLES

```
AIIP    --  INTERMEDIATE VARIABLE
AIPIP   --  INTERMEDIATE VARIABLE-A(IPIP)
```

*

```
AIPJP    --  INTERMEDIATE VARIABLE-A(IPJP)
AJPJP    --  INTERMEDIATE VARIABLE-A(JPJP)
CS       --  INTERMEDIATE VARIABLE
DEL      --  DIFFERENCE IN ELEMENTS OF A
I        --  INDEX
II       --  INDEX
IIP      --  INDEX
IJ       --  INDEX
IJP      --  INDEX
IP       --  INDEX
IPI      --  INDEX
IPIP     --  INDEX
IPJP     --  INDEX
IPP1     --  INDEX
IP1      --  INDEX
IREDO    --  COUNTER
J        --  INDEX
JP       --  INDEX
JPI      --  INDEX
JPJP     --  INDEX
KR       --  DIMENSION OF A
KRP1     --  KR + 1
NM1      --  N - 1
RAD      --  INTERMEDIATE VARIABLE
SN       --  INTERMEDIATE VARIABLE
TN       --  INTERMEDIATE VARIABLE
T1       --  LARGEST OFF-DIAGONAL ELEMENT
VIIP     --  INTERMEDIATE VARIABLE
```

27. LAMB

```
VARIABLES
 A       --  SEMI-MAJOR AXIS OF HELIOCENTRIC ELLIPSE
 E       --  ECCENTRICITY OF HELIOCENTRIC ELLIPSE
 GM      --  GRAVITATIONAL CONSTANT OF SUN
 LOC     --  FLAG INDICATING WHETHER ITERATION PROCESS
             CONVERGED OR FAILED
 NTYS    --  FLAG SPECIFYING WHETHER PSI IS LESS THAN 180
             DEGREES OR IS GREATER THAN 180 DEGRESS
 P       --  SEMI-LATUS RECTUM OF HELIOCENTRIC ELLIPSE
 PSI     --  TRANSFER ANGLE
 RL      --  HELIOCENTRIC LAUNCH PLANET RADIUS
 RP      --  HELIOCENTRIC TARGET PLANET RADIUS
 TF      --  TIME OF FLIGHT
 VL      --  TRUE ANOMALY AT LAUNCH
 VP      --  TRUE ANOMALY AT ARRIVAL
```

28. MATIN

```
VARIABLES
 AL      --  A(LL) + S (INTERMEDIATE VARIABLE)
 ALBAR   --  INTERMEDIATE VARIABLE
 B       --  INTERMEDIATE VECTOR
 DETR    --  INTERMEDIATE VECTOR
 G       --  INTERMEDIATE VECTOR
 I       --  INDEX
```

*

```
IJ      --  INDEX
IK      --  INTERMEDIATE INDEX
IL      --  INTERMEDIATE INDEX
IOFF    --  INDEX
IX      --  INTERMEDIATE VECTOR
IXL     --  INTERMEDIATE INDEX
J       --  INDEX
JI      --  INTERMEDIATE INDEX
JL      --  INTERMEDIATE INDEX
JOFF    --  INTERMEDIATE INDEX
K       --  INDEX
KJ      --  INTERMEDIATE INDEX
KJA     --  INTERMEDIATE INDEX
KR      --  DIMENSION OF A
L       --  INDEX
LI      --  INTERMEDIATE INDEX
LJ      --  INTERMEDIATE INDEX
LL      --  INDEX
L1      --  INTERMEDIATE INDEX
MIXI    --  INTERMEDIATE VARIABLE
MIXJ    --  INTERMEDIATE VARIABLE
MIXL    --  INTERMEDIATE VARIABLE
S       --  INTERMEDIATE VARIABLE
X       --  INTERMEDIATE VARIABLE
XOFF    --  INTERMEDIATE VARIABLE
```

29. MENO

```
COMMON
 CONST
 MISC
 SIMCNT
 SIM1
 STM
VARIABLES
 I       --  INDEX
 J       --  INDEX
```

30. MUND

```
COMMON
 BLK
 CONST3
 MISC
 STM
 STVEC
 TIM
 TRAJCD
 VM
VARIABLES
 I       --  INDEX
 IPR     --  TEMPORARY STORAGE FOR IPRINT
 J       --  INDEX
 RPER    --  ALTERED POSITION AND VELOCITY OF VEHICLE
             AT FINAL TIME
```

*

```
      SAVE     --   TEMPORARY STORAGE LOCATION FOR GRAVITATIONAL
                    CONSTANTS OF SUN AND TARGET PLANET
      THETA    --   AUGMENTED PORTION OF STATE TRANSITION MATRIX

31. NAVM
     COMMON
      STM
      STVEC
     VARIABLES
      A        --   INTERMEDIATE VECTOR
      I        --   INDEX
      J        --   INDEX
      K        --   INDEX
      KK       --   INDEX
      L        --   INDEX
      NM       --   INDEX
      N1       --   DIMENSION OF P
      PPHT     --   INTERMEDIATE ARRAY

32. NDTM
     COMMON
      MISC
      STM
      TIM
      TRAJCD
      VM
     VARIABLES
      F1       --   TEMPORARY STORAGE FOR FACP
      F2       --   TEMPORARY STORAGE FOR FACV
      T        --   INDEX
      II       --   INDEX
      IPR      --   INTERMEDIATE STORAGE FOR IPRINT
      M        --   INDEX
      N        --   INDEX
      RP       --   POSITION AND VELOCITY OF VEHICLE AT INITIAL
                    TIME
      SAVE     --   TEMPORARY STORAGE FOR ACC
      T        --   ALTERED POSITION AND VELOCITY OF VEHICLE AT
                    INITIAL TIME
      U        --   ALTERED POSITION AND VELOCITY OF VEHICLE AT
                    FINAL TIME

33. NEWPGE
     COMMON
      BLK
      COM
      PRT

34. NJEXN
     COMMON
      BLK
     VARIABLES
      ANG1     --   ANGLE OF FIRST BURN
      ANG2     --   ANGLE OF SECOND BURN
```

*

AU	--	CONVERSION UNIT FROM KM TO ASTRONOMICAL UNITS
AUDAY	--	CONVERSION UNIT FROM KM/SEC TO AU/DAY
AUS	--	CONVERSION UNIT FROM KM TO AU
AZI	--	AZIMUTH ANGLE AT INJECTION
CCA	--	HELIOCENTRIC SEMI-MAJOR AXIS
CCE	--	HELIOCENTRIC ECCENTRICITY
CCEA1	--	ECCENTRIC ANOMALY ON HELIOCENTRIC ELLIPSE AT LAUNCH
CCI	--	HELIOCENTRIC INCLINATION
CCL	--	HELIOCENTRIC LONGITUDE OF ASCENDING NODE
CCM1	--	MEAN ANOMALY ON HELIOCENTRIC ELLIPSE AT LAUNCH
CCN	--	NORMAL TO HELIOCENTRIC PLANE
CCP	--	HELIOCENTRIC SEMI-LATUS RECTUM
CCPSI	--	HELIOCENTRIC CENTRAL ANGLE
CCRM1	--	MAGNITUDE OF HELIOCENTRIC RADIUS AT LAUNCH
CCR1	--	HELIOCENTRIC RADIUS VECTOR AT LAUNCH (AU)
CCTA1	--	TRUE ANOMALY ON HELIOCENTRIC ELLIPSE AT LAUNCH
CCVM1	--	MAGNITUDE OF HELIOCENTRIC VELOCITY AT LAUNCH (AU/DAY)
CCV1	--	HELIOCENTRIC VELOCITY VECTOR AT LAUNCH (AU/DAY)
CCW	--	HELIOCENTRIC ARGUMENT OF PERIAPSIS
CI	--	COSINE OF INCLINATION (OF EQUATOR W.R.T. ECLIPTIC)
CL	--	COSINE OF LONGITUDE OF ASCENDING NODE (OF EQUATOR W.R.T. ECLIPTIC)
CONV	--	CONVERSION UNIT FROM KM/SEC TO AU/DAY
C3	--	ENERGY MEASURE
DDAZ	--	AZIMUTH ANGLE AT LAUNCH
DDG	--	GRAVITATIONAL CONSTANT OF LAUNCH PLANET
DDIQ	--	INCLINATION OF EQUATOR W.R.T. ECLIPTIC
DDJD	--	JULIAN DATE OF LAUNCH (IF JINJT=0, OUTPUT IS INJECTION TIME)
DDLAT	--	LATITUDE OF LAUNCH SITE
DDLON	--	LONGITUDE OF LAUNCH SITE
DDLQ	--	LONGITUDE OF ASCENDING NODE OF EQUATOR W.R.T. ECLIPTIC
DDN	--	NORMAL VECTOR TO LAUNCH-PLANET-PHASE ORBIT (EQUATORIAL COORDINATES)
DDR1	--	POSITION OF LAUNCH PLANET AT LAUNCH (HELIO-CENTRIC ECLIPTIC COORDINATES)
DDS	--	SPHERE OF INFLUENCE OF LAUNCH PLANET (KM)
DDV1	--	VELOCITY OF LAUNCH PLANET AT LAUNCH (HELIOCENTRIC ECLIPTIC COORDINATES)
DLAQ	--	DECLINATION OF DEPARTURE ASYMPTOTE W.R.T. EQUATOR
DPA	--	DECLINATION OF INCOMING ASYMPTOTE AT TARGET PLANET (W.R.T. ECLIPTIC)
ECEQ	--	TRANSFORMATION MATRIX FROM ECLIPTIC TO EQUATORIAL COORDINATES
HHA	--	LAUNCH PLANET HYPERBOLA SIMI-MAJOR AXIS

*

HHE -- LAUNCH PLANET HYPERBOLA ECCENTRICITY
HHI -- LAUNCH PLANET HYPERBOLA INCLINATION(W.R.T. EQUATOR)
HHL -- LAUNCH PLANET HYPERBOLA LONGITUDE OF ASCENDING NODE
HHN -- LAUNCH PLANET HYPERBOLA NORMAL VECTOR (EQUATORIAL COORDINATES)
HHP -- LAUNCH PLANET HYPERBOLA SEMI-LATUS RECTUM
HHPV -- ALUNCH PLANET HYPERBOLA P-VECTOR (UNIT VECTOR TOWARDS PERIAPSIS)
HHQ -- LAUNCH PLANET HYPERBOLA Q-VECTOR (UNIT VECTOR NORMAL TO P-VECTOR)
HHR1 -- INJECTION POSITION VECTOR (LAUNCH PLANET ECLIPTIC)
HHRQ1 -- POSITION VECTOR AT INJECTION (LAUNCH PLANET EQUATORIAL)
HHRQ2 -- POSITION VECTOR AT SPHERE OF INFLUENCE (LAUNCH PLANET EQUATORIAL)
HHR1 -- POSITION VECTOR AT INJECTION (LAUNCH PLANET ECLIPTIC)
HHR2 -- POSITION VECTOR AT SPHERE OF INFLUENCE (LAUNCH PLANET EQUATORIAL)
HHTA -- TRUE ANOMALY ALONG LAUNCH PLANET HYPERBOLA AT INJECTION
HHTA1 -- TRUE ANOMALY ALONG LAUNCH PLANET HYPERBOLA AT INJECTION (SHOULD EQUAL HHTA)
HHT1 -- TIME FROM PERIAPSIS TO INJECTION ALONG HYPERBOLA
HHVDC -- DECLINATION OF DEPARTURE ASYMPTOTE (ECLYPTIC COORDINATES)
HHVEQ -- DEPARTURE ASYMTOTE (VELOCITY) IN EQUATORIAL COORDINATES
HHVM1 -- SPEED AT INJECTION (W.R.T. LAUNCH PLANET)
HHVQ1 -- VECTOR POSITION AT INJECTION (LAUNCH PLANET ECLIPTIC)
HHVQ2 -- VECTOR POSITION AT SPHERE OF INFLUENCE (LAUNCH PLANET EQUATORIAL)
HHVRA -- RIGHT ASCENSION OF DEPARTURE ASYMPTOTE (ECLIPTIC COORDINATES)
HHV1 -- VELOCITY VECTOR AT INJECTION (LAUNCH PLANET ECLIPTIC)
HHV2 -- VELOCITY VECTOR AT SPHERE OF INFLUENCE LAUNCH PLANET ECLIPTIC
HHW -- ARGUMENT OF PERIAPSIS ON HYPERBOLA
IVHL -- FLAG INDICATING WHETHER DEPARTURE OR ARRIVAL-ASYMPTOTE ELEMENTS ARE BEING COMPUTED
JC3 -- FLAG INDICATING WHETHER BIASED (JC3=1) OR UNBIASED (JC3=0) CONDITIONS ARE GENERATED
JINJT -- FLAG INDICATING WHETHER INJECTION TIME IS UPDATED (JINJT=0) OR NOT (JINJT=1)
LOC -- FLAG INDICATING WHETHER CONVERGENCE OCCURRED LOC LESS THAN 4,OR NOT,LOC GREATER THAN 4
NDD -- INDEX SPECIFYING LAUNCH PLANET
NTDD -- YEAR,MONTH,DAY,HOUR,MINUTE OF LAUNCH

*

NTT	--	INDEX SPECIFYING TARGET PLANET
NTTT	--	YEAR,MONTH,DAY,HOUR,MINUTE OF ENCOUNTER
NTYS	--	FLAG SET TO 1 FOR CCPSI BETWEEN 0 AND 180 DEGREES, SET TO 2 FOR CCPSI BETWEEN 180 AND 360 DEGREES
PHI	--	INJECTION LATITUDE
PI	--	CONSTANT = 3.141591
PR	--	PARKING ORBIT RADIUS
PTH	--	INJECTION PATH ANGLE
P1	--	INTERMEDIATE VARIABLE
P2	--	INTERMEDIATE VARIABLE
RAD	--	CONVERSION UNIT FROM DEGREES TO RADIANS
RAI	--	INJECTION RIGHT ASCENSION
RALQ	--	RIGHT ASCENSION OF DEPARTURE HYPERBOLA (LAUNCH PLANET EQUATORIAL)
RAP	--	RIGHT ASCENSION OF ARRIVAL HYPERBOLA (TARGET PLANET ECLIPTIC)
RJ	--	INJECTION RADIUS
RL	--	HELIOCENTRIC RADIUS OF LAUNCH PLANET AT INITIAL TIME (AU)
ROT	--	ROTATIONAL RATE OF LAUNCH PLANET
RP	--	HELIOCENTRIC RADIUS OF TARGET PLANET AT FINAL TIME (AU)
S	--	EXCESS VELOCITY AT TARGET PLANET
SDD	--	SECONDS OF LAUNCH DATE)
SI	--	SINE OF INCLINATION OF EQUATORIAL W.R.T. ECLIPTIC
SJC3	--	STORAGE FOR JC3
SL	--	SINE OF LONGITUDE OF ASCENDING NODE OF EQUATORIAL W.R.T. ECLIPTIC
SRL	--	STORAGE FOR RL
SSG	--	GRAVITATIONAL CONSTANT OF SUN
STA1	--	STORAGE FOR CCTA1
STT	--	SECONDS OF TARGET DATE
SXL	--	STORAGE FOR XL
TB	--	TIME FROM LAUNCH TO INJECTION
TC	--	COAST TIME IN PARKING ORBIT
TF	--	TOTAL FLIGHT TIME FOR HELIOCENTRIC TRANSFER
THI	--	INJECTION LONGITUDE
TIM1	--	TIME (SEC) OF FIRST BURN
TIM2	--	TIME (SEC) OF SECOND BURN
TINJ	--	TIME (HOURS) OR DATE OF INJECTION
TTJD	--	JULIAN DATE OF ENCOUNTER
TTN	--	NORMAL TO PLANE AT TARGET PLANET (TARGET PLANET ECLIPTIC)
TTR7	--	POSITION VECTOR AT ARRIVAL AT TARGET PLANET (TARGET PLANET ECLIPTIC)
TTV7	--	VELOCITY VECTOR AT ARRIVAL AT TARGET PLANET (TARGET PLANET ECLIPTIC)
VHL	--	MAGNITUDE OF DEPARTURE ASYMPTOTE VELOCITY
VHP	--	MAGNITUDE OF ARRIVAL ASYMPTOTE VELOCITY
VL	--	TRUE ANOMALY OF HELIOCENTRIC ORBIT AT LAUNCH
VP	--	TRUE ANOMALY OF HELIOCENTRIC ORBIT AT ENCOUNTER

*

```
     XL        --  STATE (POSITION,VELOCITY) OF LAUNCH PLANET
                   AT LAUNCH (HELIOCENTRIC ECLIPTIC)
     XP        --  STATE (POSITION,VELOCITY) OF TARGET PLANET
                   AT ENCOUNTER (HELIOCENTRIC ECLIPTIC)
35. NTM
    COMMON
     BLK
     EVENT
     MISC
     SIMCNT
     SIM2
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     ACCS      --  TEMPORARY STORAGE FOR ACC
     D1        --  JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL
                   TRAJECTORY TIME
     I         --  INDEX
     K1        --  INDEX
     K2        --  INDEX
     K3        --  INDEX
     NBODS     --  TEMPORARY STORAGE FOR NBOD
     NBS       --  TEMPORARY VECTOR FOR NB
     RMP       --  DISTANCE OF VEHICLE FROM TARGET PLANET AT
                   SPHERE OF INFLUENCE OR CLOSEST APPROACH
     SAVE1     --  TEMPORARY STORAGE FOR GRAVITATIONAL CONSTANT
                   OF SUN
     SAVE10    --  TEMPORARY STORAGE FOR INCLINATION CONSTANTS
                   OF TARGET PLANET
     SAVE11    --  SAME COMMENTS AS SAVE10
     SAVE12    --  SAME COMMENTS AS SAVE10
     SAVE13    --  SAME COMMENTS AS SAVE10
     SAVE2     --  TEMPORARY STORAGE FOR GRAVITATIONAL CONSTANT
                   OF TARGET PLANET
     SAVE3     --  TEMPORARY STORAGE FOR CONSTANTS OF SEMI-
                   MAJOR AXIS OF TARGET PLANET
     SAVE4     --  SAME COMMENTS AS SAVE3
     SAVE5     --  TEMPORARY STORAGE FOR ECCENTRICITY CONSTANTS
                   OF TARGET PLANET
     SAVE6     --  SAME COMMENTS AS SAVE5
     SAVE7     --  SAME COMMENTS AS SAVE5
     SAVE8     --  SAME COMMENTS AS SAVE5
     VMP       --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
                   TARGET PLANET AT SPHERE OF INFLUENCE OR
                   CLOSEST APPROACH

36. ORB
    COMMON
     BLK
     PRT
    VARIABLES
     I         --  CODE OF PLANET
```

*

```
     IK       --  INDEX
     IL       --  INDEX
     IT       --  INDEX
     ITEMP    --  ITERMEDIATE VARIABLE
     J        --  INDEX
     K        --  INDEX
     PI2      --  TWICE THE MATHEMATICAL CONSTANT PI

37.  OT2
    VARIABLES
     AZI      --  INJECTION AZIMUTH
     C3       --  ENERGY OF HYPERBOLIC ORBIT
     CCA      --  SEIM-MAJOR AXIS OF HELIOCENTRIC CONIC
     CCE      --  HELIOCENTRIC ECCENTRICITY
     CCI      --  INCLINATION OF HELIOCENTRIC CONIC
     CCPSI    --  CENTRAL ANGLE OF HELIOCENTRIC CONIC
                  INITIAL TIME
     CCTA1    --  TRUE ANOMALY ON HELIOCENTRIC CONIC AT
     CCTA7    --  TRUE ANOMALY ON HELIOCENTRIC CONIC AT
                  FINAL TIME
     CCVM1    --  HELIOCENTRIC SPEED OF LAUNCH PLANET AT
                  INITIAL TIME
     DDAZ     --  LAUNCH AZIMUTH
     DDRM1    --  HELIOCENTRIC DISTANCE TO LAUNCH PLANET AT
                  INITIAL TIME
     DDVM1    --  SPEED ON HELIOCENTRIC CONIC AT INITIAL TIME
     DLAQ     --  DECLINATION OF HHVM2
     DPA      --  DECLINATION OF VHP
     HHE      --  ECCENTRICITY OF HYPERBOLA
     HHVM2    --  HYPERBOLIC EXCESS VELOCITY AT LAUNCH PLANET
     HHVQM    --  INJECTION VELOCITY
     NTDD     --  YEAR-MONTH-DAY-HOUR-MIN OF INITIAL TIME
     NTTT     --  YEAR-MONTH-DAY-HOUR-MIN OF FINAL TIME
     PHI      --  INJECTION LATITUDE
     PTH      --  TRAJECTORY PATH ANGLE
     RAI      --  INJECTION RIGHT ASCENSION
     RALQ     --  RIGHT ASCENSION OF HHVM2
     RAP      --  RIGHT ASCENSION OF VHP
     RJ       --  INJECTION RADIUS
     TB       --  TIME FROM LAUNCH TO INJECTION
     TC       --  COAST TIME
     TF       --  FLIGHT TIME
     THI      --  INJECTION LONGITUDE
     TINJ     --  TIME OF INJECTION ON LAUNCH DATE (HOURS)
     TL       --  TIME OF LAUNCH ON LAUNCH DATE (HOURS)
     TTRM7    --  HELIOCENTRIC DISTANCE TO TARGET PLANET AT
                  FINAL TIME
     TTVM7    --  HELIOCENTRIC SPEED OF TARGET PLANET AT FINAL
                  TIME
     VHP      --  HYPERBOLIC EXCESS VELOCITY AT TARGET PLANET
     XL       --  STATE OF LAUNCH PLANET AT INITIAL DATE
     XP       --  STATE OF TARGET PLANET AT TARGET DATE
```

*

```
38. OUT1
    VARIABLES
     BDR      --   IMPACT PLANE PARAMETER
     BDT      --   IMPACT PLANE PARAMETER
     DINCL    --   INCLINATION AT CLOSEST APPROACH
     GMS      --   GRAVITATIONAL CONSTANT OF SUN
     IDAT1    --   DATE OF INJECTION (YEAR, MONTH, DAY, HOUR,
                   MINUTE)
     IDAT2    --   DATE AT SPHERE OF INFLUENCE (YEAR, MONTH,
                   DAY, HOUR, MINUTE)
     IDAT3    --   DATE AT CLOSEST APPROACH (YEAR, MONTH, DAY,
                   HOUR, MINUTE)
     INJEK    --   INJECTION OPTION FLAG
     ITARG    --   TARGETING OPTION FLAG
     NB       --   INDICES OF GRAVITATIONAL BODIES
     NITS     --   MAXIMUM ALLOWABLE ITERATIONS AT FINAL LEVEL
     R        --   POSITION MAGNITUDE
     RCA      --   RADIUS AT CLOSEST APPROACH
     RP       --   POSITION VECTOR (HELIOCENTRIC ECLIPTIC)
     S1       --   DATE OF INJECTION (SECONDS)
     S2       --   DATE AT SPHERE OF INFLUENCE (SECONDS)
     S3       --   DATE AT CLOSEST APPROACH (SECONDS)
     TOL1     --   ACCEPTABLE TOLERANCES ON TARGET CONSTRAINTS
     TOL2     --   SAME AS TOL1
     TOL3     --   SAME AS TOL2
     V        --   SPEED
     VP       --   VELOCITY VECTOR (HELIOCENTRIC ECLIPTIC)
     WP       --   NORMAL TO PLANE

39. PARTL
    VARIABLES
     H3       --   INTERMEDIATE VARIABLE
     PBR      --   PARTIAL OF B DOT R WITH RESPECT TO STATE
                   VECTOR
     PBT      --   PARTIAL OF B DOT T WITH RESPECT TO STATE
                   VECTOR
     RV       --   INTERMEDIATE VARIABLE
     S        --   MAGNITUDE OF VELOCITY
     U        --   INTERMEDIATE VARIABLE
     UV       --   INTERMEDIATE VARIABLE
     UV3      --   CUBE OF UV
     U2       --   SQUARE OF U
     U2PV2    --   INTERMEDIATE VARIABLE
     V2       --   SQUARE OF MAGNITUDE OF VELOCITY

40. PCTM
    COMMON
     BLK
     MISC
     STM
     TIM
     VM
    VARIABLES
     D        --   JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL
                   TIME
     DELT     --   LENGTH OF TIME INCREMENT IN PROPER UNITS
```

*

DUM	--	TEMPORARY STORAGE FOR STATE TRANSITION MATRIX
GMS	--	GRAVITATIONAL CONSTANT OF GOVERNING BODY
I	--	INDEX
IP	--	CODE OF PLANET
J	--	INDEX
K	--	INDEX
RM	--	DISTANCE FROM SPECIFIED PLANET
RS	--	POSITION OF VEHICLE RELATIVE TO SPECIFIED PLANET
VS	--	VELOCITY OF VEHICLE RELATIVE TO SPECIFIED PLANET

41. PECEQ

VARIABLES

AC	--	ACCURACY LEVELS
ACC	--	FINAL ACCURACY LEVEL
D	--	JULIAN DATE (REFERENCED TO 1900)
DELTP	--	DAYS BETWEEN PRINTOUTS IN FINAL INTEGRATION
ECEQ	--	TRANSFORMATION MATRIX
INPR	--	INTEGRATION INCREMENTS BETWEEN PRINTOUTS IN FINAL INTEGRATION
ISKEJ	--	NUMBER OF ACCURACY LEVELS IN TARGETING SCHEDULE
MIDI	--	NUMBER OF ITERATIONS MADE AT INTERMEDIATE ACCURACY LEVELS
NBOD	--	NUMBER OF GRAVITATIONAL BODIES
NP	--	INDEX OF PLANET
RS	--	INJECTION STATE (POSITION AND VELOCITY VECTORS)
XI	--	THE INCLINATION OF THE ORBITAL PLANE TO THE ECLIPTIC
XL	--	THE LONGITUDE OF THE ASCENDING NODE OF THE ORBITAL PLANE TO THE ECLIPTIC

42. PLANE

VARIABLES

HCA	--	HELIOCENTRIC CENTRAL ANGLE
HCI	--	INCLINATION OF HELIOCENTRIC PLANE
HCN	--	NORMAL TO HELIOCENTRIC PLANE
HCW	--	LONGITUDE OF HELIOCENTRIC PLANE
NTYS	--	FLAG, NTYS=1 FOR HCA BETWEEN 0 AND 180 DEGREES NTYS=2 FOR HCA BETWEEN 180 AND 360 DEGREES
XL	--	INITIAL STATE (POSITION,VELOCITY) OF LAUNCH PLANET
XP	--	FINAL STATE OF TARGET PLANET

43. PLND

COMMON

BLK
CONST3
MISC
STM
STVEC

*

```
     TIM
     TRAJCD
     VM
    VARIABLES
     I         --   INDEX
     IPR       --   TEMPORARY STORAGE FOR IPRINT
     J         --   INDEX
     K         --   INDEX
     KJ        --   INDEX
     RPER      --   ALTERED FINAL POSITION AND VELOCITY OF
                    VEHICLE
     SAVE1     --   TEMPORARY STORAGE FOR CONSTANTS OF AUGMENTED
                    ELEMENTS OF TARGET PLANET
     SAVE2     --   SAME COMMENTS AS SAVE1
     SAVE3     --   SAME COMMENTS AS SAVE1
     SAVE4     --   SAME COMMENTS AS SAVE1
     THTWG     --   TEMPORARY LOCATION OF AUGMENTED PORTION OF
                    STATE TRANSITION MATRIX

44. POSVL
    VARIABLES
     A         --   SEMI-MAJOR AXIS
     AM        --   MEAN ANOMALY
     E         --   ECCENTRICITY
     W         --   ARGUMENT OF PERIAPSIS
     WC        --   LONGITUDE OF ASCENDING NODE
     XI        --   INCLINATION
     XIQ       --   THE INCLINATION OF THE PLANET EQUATION TO
                    THE ORBITAL PLANE
     XLQ       --   THE LONGITUDE OF THE ASCENDING NODE OF THE
                    PLANET EQUATOR TO THE ORBITAL PLANE

45. PRED
    COMMON
     CONST2
     EVENT
     GUI
     MISC
     NAME
     STM
     STVEC
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     DUM       --   COVARIANCE OF UNCERTAINTIES IN B DOT T AND
                    B DOT R
     DUM2      --   EIGENVECTORS
     DUM3      --   EIGENVALUES
     EGVCT     --   EIGENVECTORS
     EGVL      --   EIGENVALUES
     I         --   INDEX
     ICODE     --   INTERNAL CODE
```

*

```
IPR      --  TEMPORARY STORAGE FOR IPRINT
J        --  INDEX
K        --  INDEX
L        --  INDEX
LINES    --  LINE COUNT
MAX      --  MAXIMUM NUMBER OF LINES PER PAGE
PEIG     --  ITERMEDIATE ARRAY
P1       --  TEMPORARY STORAGE FOR P
RF       --  POSITION AND VELOCITY OF VEHICLE AT TIME OF
             PREDICTION EVENT
RHO      --  CORRELATION COEFFICIENT MATRIX
TPT      --  PREDICTION TIME
VEIG     --  INTERMEDIATE VECTOR

46. PRESIM
    COMMON
     CONST
     CONST2
     EVENT
     GUI
     MISC
     NAME
     SIM1
     SIM2
     STM
     STVEC
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     DM       --  COVARIANCE OF UNCERTAINTIES IN B DOT T AND
                  B DOT R
     DM2      --  EIGENVECTORS
     DM3      --  EIGENVALUES
     DUM      --  INTERMEDIATE VECTOR
     EGVCT    --  EIGENVECTORS
     EGVL     --  EIGENVALUES
     I        --  INDEX
     ICODE    --  INTERNAL CODE
     ICODE1   --  INTERNAL CODE
     IPR      --  TEMPORARY STORAGE FOR IPRINT
     J        --  INDEX
     K        --  INDEX
     L        --  INDEX
     LINES    --  LINE COUNT
     MAX      --  MAXIMUM NUMBER OF LINES PER PAGE
     PEIG     --  INTERMEDIATE ARRAY
     P1       --  TEMPORARY STORAGE FOR P
     RF       --  POSITION AND VELOCITY OF VEHICLE ON ORIGINAL
                  NOMINAL AT TIME OF PREDICTION EVENT
     RF1      --  POSITION AND VELOCITY OF VEHICLE ON MOST
                  RECENT NOMINAL AT TIME OF PREDICTION EVENT
     RF2      --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
                  TRAJECTORY AT TIME OF PREDICTION EVENT
```

*

```
     RHO      --  CORRELATION COEFFICIENT MATRIX
     RI2      --  POSITION AND VELOCITY OF VEHICLE ON ACTUAL
                  TRAJECTORY AT TIME OF LAST MEASUREMENT OR
                  EVENT
     TPT      --  PREDICTION TIME
     VEIG     --  INTERMEDIATE VECTOR

47. PRINT
    COMMON
     BLK
     COM
     PRT
    VARIABLES
     D        --  JULIAN DATE OF FINAL TIME
     I        --  INDEX
     IDAY     --  DAY OF CALENDAR DATE OF FINAL TIME
     IHR      --  HOUR OF CALENDAR DATE OF FINAL TIME
     INCMNT   --  TOTAL INCREMENTS
     IP       --  CODE OF PLANET
     IYR      --  YEAR OF CALENDAR DATE OF FINAL TIME
     J        --  INDEX
     K        --  INDEX
     MIN      --  MINUTES OF CALENDAR DATE OF FINAL TIME
     MO       --  MONTH OF CALENDAR DATE OF FINAL TIME
     N2       --  INDEX
     RP       --  HELIOCENTRIC RADIUS OF PLANET
     RS       --  HELIOCENTRIC RADIUS OF VEHICLE
     RV       --  HELIOCENTRIC RADIUS OF VIRTUAL MASS
     SEC      --  SECONDS OF CALENDAR DATE OF FINAL TIME
     TMP      --  POSITION AND VELOCITY OF VIRTUAL MASS RELATIVE
                  TO PLANETS
     VMR      --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
                  VIRTUAL MASS
     VP       --  MAGNITUDE OF VELOCITY OF PLANET
     VS       --  MAGNITUDE OF VELOCITY OF VEHICLE
     VSP      --  MAGNITIDE OF VELOCITY OF VEHICLE RELATIVE TO
                  PLANET
     VV       --  MAGNITUDE OF VELOCITY OF VIRTUAL MASS

48. PRINT1
    COMMON
     BLK
     MISC
     PRT
     STVEC
     TIM
     TRAJCD
     VM
    VARIABLES
     D1       --  JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL
                  TRAJECTORY TIME
     D2       --  JULIAN DATE, EPOCH JAN.0, 1900, OF FINAL
                  TRAJECTORY TIME
     D3       --  JULIAN DATE OF INITIAL TRAJECTORY TIME
```
*

```
D4      --  JULIAN DATE OF FINAL TRAJECTORY TIME
I       --  INDEX
IDAY    --  CALENDAR DAY OF FINAL TIME
IHR     --  CALENDAR HOUR OF FINAL TIME
IMIN    --  CALENDAR MINUTE OF FINAL TIME
IMO     --  CALENDAR MONTH OF FINAL TIME
IYR     --  CALENDAR YEAR OF FINAL TIME
LDAY    --  CALENDAR DAY OF INITIAL TIME
LHR     --  CALENDAR HOUR OF INITIAL TIME
LMIN    -- CALENDAR MINUTE OF INITIAL TIME
LMO     --  CALENDAR MONTH OF INITIAL TIME
LYR     --  CALENDAR YEAR OF INITIAL TIME
MAX     --  MAXIMUM NUMBER OF LINES PER PAGE
RI      --  POSITION AND VELOCITY OF VEHICLE AT INITIAL
            TIME
RMF     --  HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME
RMI     --  HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL
            TIME
SECI    --  CALENDAR SECONDS OF FINAL TIME
SECL    --  CALENDAR SECONDS OF INITIAL TIME
TRTM2   --  FINAL TRAJECTORY TIME
VE      --  POAITION AND VELOCITY OF VEHICLE RELATIVE TO
            EARTH AT FINAL TIME
VMF     --  MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL
            TIME
VMI     --  MAGNITUDE OF VELOCITY OF VEHICLE AT INITIAL
            TIME
VT      --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
            TARGET PLANET AT FINAL TIME

49. PRINT3
    COMMON
     BLK
     MEAS
     MISC
     NAME
     STM
     STVEC
     TIM
     VM
    VARIABLES
     D       --  INTERMEDIATE DATE
     D1      --  JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL
                 TRAJECTORY TIME
     D2      --  JULIAN DATE, EPOCH JAN.0,1900, OF FINAL TIME
     D3      --  JULIAN DATE OF INITIAL TIME
     D4      --  JULIAN DATE OF FINAL TIME
     I       --  INDEX
     IA      --  STATION NUMBER
     IDAY    --  CALENDAR DAY OF FINAL TIME
     IHR     --  CALENDAR HOUR OF FINAL TIME
     IMIN    --  CALENDAR MINUTE OF FINAL TIME
     IMO     --  CALENDAR MONTH OF FINAL TIME
     ITEMP   --  INTERMEDIATE VARIABLE
```

*

```
    IYR     --  CALENDAR YEAR OF FINAL TIME
    J       --  INDEX
    LDAY    --  CALENDAR DAY OF INITIAL TIME
    LHR     --  CALENDAR HOUR OF INITIAL TIME
    LINES   --  LINE COUNT
    LMIN    --  CALENDAR MINUTES OF INITIAL TIME
    LMO     --  CALENDAR MONTH OF INITIAL TIME
    LYR     --  CALENDAR YEAR OF INITIAL TIME
    M       --  NUMBER OF MEASUREMENT
    MAX     --  MAXIMUM NUMBER OF LINES PER PAGE
    RME     --  GEOCENTRIC RADIUS OF VEHICLE
    RMP     --  DISTANCE OF VEHICLE FROM TARGET PLANET
    SECI    --  CALENDAR SECONDS OF FINAL TIME
    SECL    --  CALENDAR SECONDS OF INITIAL TIME
    TRTM2   --  TRAJECTORY TIME AT END OF INTERVAL
    VME     --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
                EARTH
    VMP     --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
                TARGET PLANET

50. PRINT4
   COMMON
    BLK
    MEAS
    MISC
    NAME
    SIMCNT
    SIM1
    SIM2
    STM
    STVEC
    TIM
    VM
   VARIABLES
    ADON    --  ACTUAL DEVIATION OF STATE VECTOR FROM ORIGINAL
                NOMINAL
    AODI    --  ACTUAL ORBIT DETERMINATION INACCURACY
    D       --  INTERMEDIATE DATE
    D1      --  JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL TIME
    D2      --  JULIAN DATE, EPOCH JAN.0,1900, OF FINAL TIME
    D3      --  JULIAN DATE OF INITIAL TIME
    D4      --  JULIAN DATE OF FINAL TIME
    EDON    --  ESTIMATED DEVIATION OF STATE VECTOR FROM
                ORIGINAL NOMINAL
    I       --  INDEX
    IA      --  STATION NUMBER
    ICODE   --  INTERNAL CODE
    IDAY    --  CALENDAR DAY OF FINAL TIME
    IHR     --  CALENDAR HOUR OF FINAL TIME
    IMIN    --  CALENDAR MINUTES OF FINAL TIME
    IMO     --  CALENDAR MONTH OF FINAL TIME
    ITEMP   --  INTERMEDIATE VARIABLE
    IYR     --  CALENDAR YEAR OF FINAL TIME
    J       --  INDEX
```

*

LDAY	--	CALENDAR DAY OF INITIAL TIME
LHR	--	CALENDAR HOURS OF INITIAL TIME
LINES	--	LINE COUNT
LMIN	--	CALENDAR MINUTES OF FINAL TIME
LMO	--	CALENDAR MONTH OF FINAL TIME
LYR	--	CALENDAR YEAR OF FINAL TIME
M	--	NUMBER OF MEASUREMENT
MAX	--	MAXIMUM NUMBER OF LINES PER PAGE
RME1	--	DISTANCE OF VEHICLE FROM EARTH ON ORIGINAL NOMINAL
RME2	--	DISTANCE OF VEHICLE FROM EARTH ON MOST RECENT NOMINAL
RME3	--	DISTANCE OF VEHICLE FROM EARTH ON ACTUAL TRAJECTORY
RMP1	--	DISTANCE OF VEHICLE FROM TARGET PLANET ON ORIGINAL NOMINAL
RMP2	--	DISTANCE OF VEHICLE FROM TARGET PLANET ON MOST RECENT NOMINAL
RMP3	--	DISTANCE OF VEHICLE FROM TARGET PLANET ON ACTUAL TRAJECTORY
SECI	--	CALENDAR SECONDS OF FINAL TIME
SECL	--	CALENDAR SECONDS OF INITIAL TIME
TRTM2	--	TRAJECTORY TIME AT END OF INTERVAL
VME1	--	VELOCITY OF VEHICLE RELATIVE TO EARTH ON ORIGINAL NOMINAL
VME2	--	VELOCITY OF VEHICLE RELATIVE TO EARTH ON MOST RECENT NOMINAL
VME3	--	VELOCITY OF VEHICLE RELATIVE TO EARTH ON ACTUAL TRAJECTORY
VMP1	--	VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ORIGINAL NOMINAL
VMP2	--	VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON MOST RECENT NOMINAL
VMP3	--	VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ACTUAL TRAJECTORY
XE	--	POSITION AND VELOCITY OF EARTH
XVE1	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO EARTH ON ORIGINAL NOMINAL
XVE2	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO EARTH ON MOST RECENT NOMINAL
XVE3	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO EARTH ON ACTUAL TRAJECTORY
XVP1	--	POITION AND VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ORIGINAL NOMINAL
XVP2	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON MOST RECENT NOMINAL
XVP3	--	POSITION AND VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ACTUAL TRAJECTORY

51. PRNTS3

COMMON

CONST

CONST2

CONST3

*

```
 EVENT
 GUI
 MEAS
 MISC
 NAME
 STM
 STVEC
 TIM
 TRAJCD
 VM
VARIABLES
 D1       --  JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL
              TIME
 D2       --  JULIAN DATE, EPOCH JAN. 0,1900, OF FINAL TIME
 D3       --  JULIAN DATE OF INITIAL TIME
 D4       --  JULIAN DATE OF FINAL TIME
 I        --  INDEX
 IDAY     --  CALENDAR DAY OF FINAL TIME
 IHR      --  CALENDAR HOUR OF FINAL TIME
 IMIN     --  CALENDAR MINUTES OF FINAL TIME
 IMO      --  CALENDAR MONTH OF FINAL TIME
 IYR      --  CALENDAR YEAR OF FINAL TIME
 J        --  INDEX
 K        --  INDEX
 LDAY     --  CALENDAR DAY OF INITIAL TIME
 LHR      --  CALENDAR HOUR OF INITIAL TIME
 LINES    --  LINE COUNT
 LMIN     --  CALENDAR MINUTES OF INITIAL TIME
 LMO      --  CALENDAR MONTH OF INITIAL TIME
 LYR      --  CALENDAR YEAR OF INITIAL TIME
 MAX      --  MAXIMUM LINES PER PAGE
 RI       --  POSITION AND VELOCITY OF VEHICLE AT INITIAL
              TIME
 RMF      --  HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME
 RMI      --  HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL
              TIME
 SECI     --  CALENDAR SECONDS OF FINAL TIME
 SECL     --  CALENDAR SECONDS OF INITIAL TIME
 TRTM2    --  TRAJECTORY TIME AT END OF TRAJECTORY
 VE       --  POSITION AND VELOCITY OF VEHICLE RELATIVE
              TO EARTH AT FINAL TIME
 VMF      --  MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL
              TIME
 VMI      --  MAGNITUDE OF VELOCITY OF VEHICLE AT INITIAL
              TIME
 VT       --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
              TARGET PLANET AT FINAL TIME

52. PRNTS4
     COMMON
      BLK
      CONST
      CONST2
      CONST3
```

*

```
 EVENT
 GUI
 MEAS
 MISC
 NAME
 PRT
 SIMCNT
 SIM1
 SIM2
 STM
 STVEC
 TIM
 TRAJCD
 TRJ
 VM
VARIABLES
 ADON     --  ACTUAL DEVIATION OF STATE VECTOR FROM ORIGINAL
              NOMINAL
 AODI     --  ACTUAL ORBIT DETERMINATION INACCURACY
 BLANK    --  BLANK HOLLERITH CHARACTER
 D1       --  JULIAN DATE, EPOCH JAN.0,1900, OF INITIAL
              TIME
 D2       --  JULIAN DATE, EPOCH JAN.0,1900, OF FINAL TIME
 D3       --  JULIAN DATE OF INITIAL TIME
 D4       --  JULIAN DATE OF FINAL TIME
 EDON     --  ESTIMATED DEVIATION OF STATE VECTOR FROM
              ORIGINAL NOMINAL
 I        --  INDEX
 IDAY     --  CALENDAR DAY OF FINAL TIME
 IHR      --  CALENDAR HOUR OF FINAL TIME
 IMIN     --  CALENDAR MINUTES OF FINAL TIME
 IMO      --  CALENDAR MONTH OF FINAL TIME
 IYR      --  CALENDAR YEAR OF FINAL TIME
 J        --  INDEX
 J1       --  INDEX
 J2       --  INDEX
 K        --  INDEX
 LDAY     --  CALENDAR DAY OF INITIAL TIME
 LHR      --  CALENDAR HOURS OF INITIAL TIME
 LINES    --  LINE COUNT
 LMIN     --  CALENDAR MINUTES OF INITIAL TIME
 LMO      --  CALENDAR MONTH OF INITIAL TIME
 LYR      --  CALENDAR YEAR OF INITIAL TIME
 MAX      --  MAXIMUM NUMBER OF LINES PER PAGE
 RE1      --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
              EARTH ON ORIGINAL NOMINAL
 RE2      --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
              EARTH ON MOST RECENT NOMINAL
 RE3      --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
              EARTH ON ACTUAL TRAJECTORY
 RI       --  POSITION AND VELOCITY OF VEHICLE ON ORIGINAL
              NOMINAL AT INITIAL TIME
 RI1      --  POSITION AND VELOCITY OF VEHICLE ON MOST
              RECENT NOMINAL AT INITIAL TIME
```

*

RME -- GEOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RME1 -- GEOCENTRIC RADIUS OF VEHICLE ON ORIGINAL NOMINAL AT FINAL TIME
RME2 -- GEOCENTRIC RADIUS OF VEHICLE ON MOST RECENT NOMINAL AT FINAL TIME
RME3 -- GEOCENTRIC RADIUS OF VEHICLE ON ACTUAL TRAJECTORY AT FINAL TIME
RMP -- DISTANCE OF VEHICLE FROM TARGET PLANET AT INITIAL TIME
RMP1 -- DISTANCE OF VEHICLE FROM TARGET PLANET ON ORIGINAL NOMINAL AT FINAL TIME
RMP2 -- DISTANCE OF VEHICLE FROM TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME
RMP3 -- DISTANCE OF VEHICLE FROM TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME
RMS -- HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RMS1 -- HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON ORIGINAL NOMINAL
RMS2 -- HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL
RMS3 -- HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON ACTUAL TRAJECTORY
RP1 -- STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON ORIGINAL NOMINAL
RP2 -- STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON MOST RECENT NOMINAL
RP3 -- STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON ACTUAL TRAJECTORY
SECI -- CALENDAR SECONDS AT FINAL TIME
SECL -- CALENDAR SECONDS AT INITIAL TIME
VME -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH AT INITIAL TIME
VME1 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON ORIGINAL NOMINAL AT FINAL TIME
VME2 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON MOST RECENT NOMINAL AT FINAL TIME
VME3 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON ACTUAL TRAJECTORY AT FINAL TIME
VMP -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT INITIAL TIME
VMP1 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ORIGINAL NOMINAL AT FINAL TIME
VMP2 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME
VMP3 -- MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME
VMS -- MAGNITUDE OF VELOCITY OF VEHICLE AT INITIAL TIME
VMS1 -- MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON ORIGINAL NOMINAL
VMS2 -- MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL TRAJECTORY

*

```
     VMS3      --  MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL
                   TIME ON ACTUAL TRAJECTORY

53. PSIM
    COMMON
     BLK
     CONST3
     MISC
     STM
     STVEC
     TIM
     TRAJCD
     VM
    VARIABLES
     D         --  INTERMEDIATE JULIAN DATE
     DELT      --  TIME INTERVAL IN CORRECT UNITS
     DUM       --  TEMPORARY STORAGE FOR STATE TRANSITION MATRIX
     I         --  INDEX
     J         --  INDEX
     POSS      --  DISTANCE OF THE VEHICLE FROM THE TARGET
                   PLANET AT INITIAL TIME
     RS        --  POSITION OF VEHICLE RELATIVE TO GOVERNING
                   BODY AT INITIAL TIME
     THSP      --  CONSTANT EQUAL TO SIX TIMES THE SPHERE OF
                   INFLUENCE OF TARGET PLANET
     VEC       --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
                   TARGET PLANET AT INITIAL TIME
     VS        --  VELOCITY OF VEHICLE RELATIVE TO GOVERNING
                   BODY AT INITIAL TIME

54. QUASI
    COMMON
     CONST
     CONST2
     EVENT
     MISC
     NAME
     SIM1
     SIM2
     STM
     STVEC
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     DUM       --  INTERMEDIATE VECTOR
     I         --  INDEX
     J         --  INDEX
     K         --  INDEX
     LINES     --  LINE COUNT
     MAX       --  MAXIMUM NUMBER OF LINES PER PAGE
     RF        --  STATE OF VEHICLE AT TIME OF QUASI-LINEAR
                   FILTERING EVENT ON ORIGINAL NOMINAL
     RF1       --  STATE OF VEHICLE AT TIME OF QUASI-LINEAR
                   FILTERING EVENT ON MOST RECENT NOMINAL
```

*

```
    RF2      --  STATE OF VEHICLE AT TIME OF QUASI-LINEAR
                 FILTERING EVENT ON ACTUAL TRAJECTORY
    RHO      --  CORRELATION COEFFICIENT MATRIX
    RI2      --  STATE OF THE VEHICLE AT TIME OF LAST
                 MEASUREMENT OR EVENT ON ACTUAL TRAJECTORY

55. RNUM
   VARIABLES
    A        --  SUM OF TWELVE RANDOM NUMBERS BETWEEN ZERO
                 AND ONE
    I        --  INDEX
    N        --  INTERMEDIATE INTEGER
    NX       --  CONTROLLING INTEGER
    Q        --  INTERMEDIATE VARIABLE
    RNUM     --  RANDOM NUMBER FROM NORMAL DISTRIBUTION WITH
                 MEAN ZERO AND STANDARD DEVIATION SIGMA
    RR       --  INTERMEDIATE VARIABLE
    SS       --  INTERMEDIATE VARIABLE
    WW       --  INTERMEDIATE VARIABLE
    W1       --  INTERMEDIATE VARIABLE
    YY       --  INTERMEDIATE VARIABLE
    Y1       --  INTERMEDIATE VARIABLE
    ZZ       --  INTERMEDIATE VARIABLE
    Z1       --  INTERMEDIATE VARIABLE

56. SCHED
   COMMON
    MEAS
   VARIABLES
    M        --  INDEX

57. SPACE
   COMMON
    BLK
    COM
    PRT

58. STAPARL
   COMMON
    CONST
    VM
   VARIABLES
    G1       --  SINE OF LATITUDE
    G2       --  COSINE OF LATITUDE
    G3       --  SINE OF LONGITUDE + CONSTANT
    G4       --  COSINE OF LONGITUDE + CONSTANT
    G5       --  SINE OF OBLIQUITY OF EARTH
    G6       --  COSINE OF OBLIQUITY OF EARTH
    OMEG     --  OMEGA IN PROPER UNITS

59. TIME
    IA       --  NUMBER OF CENTURIES
    IB       --  YEARS IN PRESENT CENTURY
    ICODE    --  INTERNAL CODE USED TO DETERMINE WHICH DATE
                 IS TO BE RETURNED--JULIAN OR CALENDAR
```

*

```
        IP       --  NUMBER OF MONTH (BASED ON MARCH AS NUMBER 0)
        IQ       --  NUMBER OF YEARS
        IR       --  NUMBER OF CENTURIES DIVIDED BY 4
        IS       --  NUMBER OF YEARS SINCE LAST 400 YEAR SECTION
                     BEGAN
        IT       --  NUMBER OF LEAP YEARS IN PRESENT CENTURY
        IU       --  NUMBER OF YEARS SINCE LAST LEAP YEAR
        IV       --  NUMBER OF DAYS IN LAST YEAR
        IX       --  INTERMEDIATE INTEGER
        J        --  INTERMEDIATE INTEGER
        JD       --  NUMBER OF DAYS IN JULIAN DATE
        P        --  JULIAN DATE
        R        --  FRACTIONAL PORTION OF DAY IN JULIAN DATE
        SEC      --  NUMBER OF SECONDS IN CALENDAR DATE

60. TRAKM
    COMMON
        BLK
        CONST
        CONST2
        CONST3
        SIMCNT
        STM
        STVEC
        TIM
        VM
    VARIABLES
        AD1      --  INTERMEDIATE VARIABLE
        AD2      --  INTERMEDIATE VARIABLE
        AD3      --  INTERMEDIATE VARIABLE
        AL       --  ALTITUDE
        ALAT     --  LATITUDE
        ALON     --  LONGITUDE
        A1       --  PARTIAL OF RANGE WITH RESPECT TO X
        A2       --  PARTIAL OF RANGE WITH RESPECT TO Y
        A3       --  PARTIAL OF RANGE WITH RESPECT TO Z
        B1       --  PARTIAL OF RANGE-RATE WITH RESPECT TO X
        B2       --  PARTIAL OF RANGE-RATE WITH RESPECT TO Y
        B3       --  PARTIAL OF RANGE-RATE WITH RESPECT TO Z
        CE       --  COSINE OF OBLIQUITY OF EARTH
        COAL1    --  COSINE OF ANGLE 1
        COAL2    --  COSINE OF ANGLE 2
        COAL3    --  COSINE OF ANGLE 3
        CP       --  COSINE OF LONGITUDE + CONSTANT
        D        --  INTERMEDIATE DATE
        DENOM    --  INTERMEDIATE VARIABLE
        E1       --  PARTIAL OF RANGE WITH RESPECT TO ALTITUDE
        E2       --  PARTIAL OF RANGE WITH RESPECT TO LATITUDE
        E3       --  PARTIAL OF RANGE WITH RESPECT TO LONGITUDE
        GECS     --  GEOCENTRIC EQUATORIAL COORDINATES OF STATION
        GELS     --  GEOCENTRIC ECLIPTIC COORDINATES OF STATION
        HECE     --  COORDINATES OF EARTH
        HECP     --  COORDINATES OF TARGET PLANET
        I        --  INDEX
```

*

IA -- STATION NUMBER
IN -- INDEX
J -- INDEX
PA -- PARTIALS
PAT1 -- INTERMEDIATE VARIABLE
PAT2 -- INTERMEDIATE VARIABLE
RADNTP -- RADIUS OF TARGET PLANET
RH -- DISTANCE OF VEHICLE FROM TARGET PLANET
RHO -- DISTANCE OF VEHICLE FROM TARGET PLANET
RHO2 -- INTERMEDIATE VARIABLE
RH2 -- INTERMEDIATE VARIABLE
RRATE -- RANGE-RATE
R1 -- RANGE
R2 -- SQUARE OF RANGE
SE -- SINE OF OBLIQUITY OF EARTH
SIAL1 -- SINE OF ANGLE 1
SIAL2 -- SINE OF ANGLE 2
SIAL3 -- SINE OF ANGLE 3
SP -- SINE OF LONGITUDE + CONSTANT
S11 -- INTERMEDIATE VARIABLE
S12 -- INTERMEDIATE VARIABLE
S13 -- INTERMEDIATE VARIABLE
S21 -- INTERMEDIATE VARIABLE
S22 -- INTERMEDIATE VARIABLE
S23 -- INTERMEDIATE VARIABLE
S31 -- INTERMEDIATE VARIABLE
S32 -- INTERMEDIATE VARIABLE
S33 -- INTERMEDIATE VARIABLE
VEC -- INTERMEDIATE VECTOR

61. TRANS
VARIABLES
CE -- COSINE OF OBLIQUITY OF EARTH
DUM -- INTERMEDIATE VARIABLE
EPS -- OBLIQUITY OF EARTH
ICODE2 -- INTERNAL CODE
SE -- SINE OF OBLIQUITY OF EARTH

62. VARADA
COMMON
BLK
MISC
TIM
TRAJCD
VM
VARIABLES
BDR1 -- TEMPORARY STORAGE FOR BDR
BDT1 -- TEMPORARY STORAGE FOR BDT
B1 -- TEMPORARY STORAGE FOR B
DSI1 -- TEMPORARY STORAGE FOR DSI
I -- INDEX
IPO -- TEMPORARY STORAGE FOR IPRINT
ISP -- TEMPORARY STORAGE FOR ISP2
N -- INDEX

*

```
    RF        --  ALTERED FINAL STATE OF VEHICLE
    XC        --  ALTERED INITIAL STATE OF VEHICLE

63. VARSIM
    COMMON
     CONST
     CONST2
     EVENT
     GUI
     MISC
     NAME
     SIM1
     SIM2
     STM
     STVEC
     TIM
     TRAJCD
     TRJ
     VM
    VARIABLES
     BDRS     --  TEMPORARY STORAGE FOR BDR
     BDTS     --  TEMPORARY STORAGE FOR BDT
     BS       --  TEMPORARY STORAGE FOR B
     I        --  INDEX
     IPR      --  TEMPORARY STORAGE FOR IPRINT
     ISPS     --  TEMPORARY STORAGE FOR ISP2
     N        --  INDEX
     RF1      --  ALTERED FINAL STATE OF VEHICLE ON MOST RECENT
                  NOMINAL
     RSIS     --  TEMPORARY STORAGE FOR RSI
     TSI1     --  TEMPORARY STORAGE FOR TSI
     VSIS     --  TEMPORARY STORAGE FOR VSI
     XC       --  ALTERED INITIAL STATE OF VEHICLE ON MOST
                  RECENT NOMINAL

64. VECTOR
    COMMON
     BLK
     COM
     PRT
    VARIABLES
     DUM      --  INTERMEDIATE VARIABLE
     I        --  INDEX
     IP3      --  INTERMEDIATE INDEX

65. VMASS
    COMMON
     BLK
     COM
     PRT
    VARIABLES
     I        --  INDEX
     IP1      --  INTERMEDIATE INDEX
     IP2      --  INTERMEDIATE INDEX
```

*

```
     IP3      --  INTERMEDIATE INDEX
     J        --  INDEX
     JP1      --  INTERMEDIATE INDEX

66. VMP
    COMMON
     BLK
     COM
     PRT
     VM
    VARIABLES
     D        --  INTERMEDIATE DATE
     DELR     --  INTERMEDIATE VARIABLE
     DELT     --  INTERMEDIATE TIME
     I        --  INDEX
     IDAY     --  CALENDAR DAY
     IHR      --  CALENDAR HOUR
     IMO      --  CALENDAR MONTH
     IP       --  NUMBER OF PLANET
     IYR      --  CALENDAR YEAR
     J        --  INDEX
     JJ       --  COUNTER
     MIN      --  CALENDAR MINUTES
     NTPI     --  INDEX
     RCM      --  MAGNITUDE OF POSITION OF VEHICLE RELATIVE TO
                  TARGET PLANET AT CLOSEST APPROACH
     RCM1     --  INITIAL RADIUS OF VEHICLE RELATIVE TO TARGET
                  PLANET
     RCM2     --  PRESENT RADIUS OF VEHICLE RELATIVE TO TARGET
                  PLANET
     SEC      --  CALENDAR SECONDS
     TIMIN    --  TOTAL TIME
     TIM1     --  CP TIME USED AT BEGINNING OF TRAJECTORY
     TIM2     --  CP TIME USED AT END OF TRAJECTORY
     TP       --  INTERMEDIATE VARIABLE
     TTG      --  GRAVITATIONAL CONSTANT OF TARGET PLANET IN
                  PROPER UNITS
     VCM      --  MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO
                  TARGET PLANET AT CLOSEST APPROACH
```

VII. PROGRAM LISTING

THIS CHAPTER CONTAINS A LISTING OF ALL SUBROUTINES DESCRIBED IN CHAPTER V IN ADDITION TO A LISTING OF THE MAIN PROGRAMS OF BOTH STEAP AND THAT USED FOR THE TARGETING MODE.

```
      PROGRAM STEAP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON /MEAS/ TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DIMENSION RI(6),RF(6)
      DIMENSION DUM(17)
      DIMENSION RI1(6),RF1(6),RI2(6),RF2(6),XF2(17),BVAL(4)
      IRUN =0
      READ (5,1000) IRUNX
 1000 FORMAT(I10)
 10   IRUN=IRUN+1
      IF(IRUN-IRUNX) 20,20,999
 20   CALL DATA
      GO TO (30,100,200,300),ITR
C
C     TRAJECTORY MODE
C
 30   DO 40 I=1,6
 40   RI(I)=XI(I)
 50   CALL NTM(RI,RF,NTMC,1)
      TRTM2=TRTM1+DELTM
      IF(TRTM2.GE.FNTM) GO TO 70
      TRTM1=TRTM2
      DO 60 I=1,6
 60   RI(I)=RF(I)
      GO TO 50
 70   CALL PRINT1(RF)
      GO TO 500
```

```
C
C       TARGETING MODE
C
100     WRITE(6,1001)
1001    FORMAT(//8X*TARGETING MODE IS NOT SUPPLIED WITH THIS DECK*)
        GO TO 500
C
C       ERROR ANALYSIS MODE
C
200     NEVENT = 1
220     DO 210I=1,6
210     RI(I)=XI(I)
230     CALL SCHED(TRTM1,TRTM2,MMCODE)
240     DELTM=TRTM2-TRTM1
        IF(TRTM2-TEV(NEVENT)) 250,250,290
250     CALL NTM(RI,RF,NTMC,1)
        DO 251 I=1,6
251     XF(I)=RF(I)
        MCNTR=MCNTR+1
        CALL PSIM(RI,RF,ISTMC)
260     CALL DYNO(0)
270     CALL TRAKM(RF,MMCODE,NR,0,AY)
        CALL MENO(MMCODE,0)
        CALL NAVM(NR,0)
        CALL PRINT3(MMCODE,NR)
280     DO 281 I=1,6
281     XI(I)=XF(I)
        TRTM1=TRTM2
        GO TO 296
290     ICODE=IEVNT(NEVENT)
        TEVN=TEV(NEVENT)
        GO TO (291,292,293), ICODE
291     CALL EIGEN(RI,TEVN)
        GO TO 295
292     CALL PRED(RI,TEVN)
        GO TO 295
293     CALL GUIDM(RI,TEVN)
295     NEVENT = NEVENT+1
296     IF(TRTM1.GE.FNTM) GO TO 294
        IF(MCNTR.LE.NMN) GO TO 220
        IF(NEVENT.LE.NEV) GO TO 290
        DELTM=FNTM-TRTM1
        DO 297 I=1,6
297     RI(I)=XI(I)
        CALL NTM(RI,RF,NTMC,1)
        DO 298 I=1,6
298     XF(I)=RF(I)
        CALL PSIM(RI,RF,ISTMC)
        CALL DYNO(0)
        CALL NAVM(1,1)
294     CALL PRNTS3(RF)
        GO TO 500
C
C       SIMULATION MODE
C
300     NEVENT=1
320     DO 321 I=1,6
        RI(I)=XI(I)
321     RI1(I)=XI1(I)
        CALL SCHED(TRTM1,TRTM2,MMCODE)
```

```
      DELTM=TRTM2-TRTM1
      IF(TRTM2-TEV(NEVENT)) 330,330,390
330   CALL NTM(RI,RF,NTMC,1)
      DO 331 I=1,6
331   XF(I)=RF(I)
      IF (NQE.NE.0) GO TO 340
      DO 332 I=1,NDIM
332   XF1(I)=XF(I)
      DO 333 I=1,6
333   RF1(I)=RF(I)
      GO TO 350
340   CALL NTM(RI1,RF1,NTMC,2)
      DO 341 I=1,6
341   XF1(I)=RF1(I)
350   MCNTR=MCNTR+1
      CALL PSIM(RI1,RF1,ISTMC)
      CALL DYNO(0)
      CALL TRAKM(RF1,MMCODE,NR,0,AY)
      CALL MENO(MMCODE,0)
      CALL NAVM(NR,0)
      DO 351 I=1,6
351   RI2(I)=XI1(I)+ADEVX(I)
      DO 353 I=1,NDIM
353   ZI(I)=XI1(I)+ADEVX(I)
      CALL NTM(RI2,RF2,NTMC,3)
      DO 352 I=1,6
352   Z(I)=RF2(I)
      CALL DYNO(1)
      DO 360 I=1,6
360   ADEVX(I)=Z(I)+W(I)-XF1(I)
      CALL TRAKM(RF1,MMCODE,NR,1,EY)
      DO 361 I=1,NDIM
361   XF2(I)=XF1(I)+ADEVX(I)
      DO 363 I=1,6
363   RF2(I)=XF2(I)
      CALL TRAKM(RF2,MMCODE,NR,2,AY)
      CALL MENO(MMCODE,1)
      DO 362 I=1,NR
362   ANOIS(I)=RNUM(SQRT(AR(I,I)))
      CALL BIAS(MMCODE,BVAL)
      DO 370 I=1,NR
370   AY(I)=AY(I)+ANOIS(I)+BVAL(I)
      DO 371 K=1,NR
      DO 371 I=1,NDIM
      DO 371 J=1,NDIM
371   EY(K)=EY(K)+H(K,I)*PSI(I,J)*EDEVX(J)
      DO 372 I=1,NR
372   RES(I)=AY(I)-EY(I)
      DO 373 I=1,NDIM
      RI1(I)=0.
      DO 373 J=1,NDIM
373   RI1(I)=RI1(I)+PSI(I,J)*EDEVX(J)
      DO 374 I=1,NDIM
      EDEVX(I)=RI1(I)
      DO 374 J=1,NR
374   EDEVX(I)=EDEVX(I)+AK(I,J)*RES(J)
      CALL PRINT4(MMCODE,NR)
      DO 380 I=1,NDIM
      XI(I)=XF(I)
380   XI1(I)=XF1(I)
```

```
      TRTM1=TRTM2
      GO TO 400
  390 ICODE=IEVNT(NEVENT)
      TEVN=TEV(NEVENT)
      GO TO (391,392,393,394), ICODE
  391 CALL EIGSIM(RI,TEVN,RI1)
      GO TO 395
  392 CALL PRESIM(RI,TEVN,RI1)
      GO TO 395
  393 CALL GUISIM(RI,TEVN,RI1)
      GO TO 395
  394 CALL QUASI(RI,TEVN,RI1)
  395 NEVENT=NEVENT+1
  400 IF(TRTM1.GE.FNTM) GO TO 440
      IF(MCNTR.LE.NMN) GO TO 320
      IF(NEVENT.LE.NEV) GO TO 390
      DELTM=FNTM-TRTM1
      DO 401 I=1,6
      RI(I)=XI(I)
  401 RI1(I)=XI1(I)
      CALL NTM(RI,RF,NTMC,1)
      DO 402 I=1,6
  402 XF(I)=RF(I)
      IF(NQE.NE.0) GO TO 410
      DO 403 I=1,NDIM
  403 XF1(I)=XF(I)
      DO 404 I=1,6
  404 RF1(I)=RF(I)
      GO TO 420
  410 CALL NTM(RI1,RF1,NTMC,2)
      DO 411 I=1,6
  411 XF1(I)=RF1(I)
  420 CALL PSIM(RI1,RF1,ISTMC)
      CALL DYNO(0)
      CALL NAVM(1,1)
      DO 421 I=1,6
  421 RI2(I)=XI1(I)+ADEVX(I)
      CALL NTM(RI2,RF2,NTMC,3)
      DO 422 I=1,6
  422 Z(I)=RF2(I)
      CALL DYNO(1)
      DO 430 I=1,6
  430 ADEVX(I)=Z(I)+W(I)-XF1(I)
      DO 431 I=1,NDIM
      DUM(I)=0.
      DO 431 J=1,NDIM
  431 DUM(I)=DUM(I)+PSI(I,J)*EDEVX(J)
      DO 432 I=1,NDIM
  432 EDEVX(I)=DUM(I)
  440 CALL PRNTS4(RF,RF1)
      GO TO 500
  500 GO TO 10
  999 CALL EXIT
      END
```

(TARGETING)

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     THE ENTIRE INPUT REQUIRED BY THE TARGETING PROGRAM IS SUPPLIED IN
C     SIX CARDS WITH EACH INDIVIDUAL CARD CONTAINING UNIFIED DATA.  THE
C     (SEQUENTIAL) CARDS AND THEIR REQUISITE FORMAT ARE LISTED BELOW.
C
C        CARD 1 -- IDAT1(5),S1,IDAT2(5),S2
C           FORMAT (I5,4I3,F7.3,5X,I4,4I3,F7.3)
C        CARD 2 -- NBOD, NB(NBOD)
C           FORMAT (I2,3X,11I5)
C        CARD 3 -- INJEK, RS(6)
C           FORMAT (I2,2X,3E15.8,3E10.3)
C        CARD 4 -- ITARG, TARG1, TARG2, TOL1, TOL2, TOL3
C           FORMAT (I2,2X,5F15.5)
C        CARD 5 -- ISKEJ, AC(ISKEJ)
C           FORMAT (I2,6X,7F10.8)
C        CARD 6 -- NITS, INCPR, TIMPR, BDELV
C           FORMAT (I2,7X,I5,5X,F9.4,F11.8)
C
C     THE DEFINITIONS OF THE ABOVE DATA ARE SUMMARIZED BELOW.
C
C        IDAT1,S1 -  THE INITIAL TIME.  IDAT1 IS A 5-VECTOR COMPOSED OF
C                    THE INITIAL YEAR,MONTH,DAY,HOUR,AND MINUTE.  S1 DE-
C                    NOTES THE SECONDS.  IF INJEK=1, THIS TIME IS SPECI-
C                    FIED ONLY TO THE DAY. IF INJEK=2, THE TIME SHOULD
C                    BE PRESCRIBED TO THE NEAREST THOUSANDTH-SECOND.
C        IDAT2,S2 -  THE TARGET TIME.  IF ITARG=1,2,5,6 THIS IS THE TIME
C                    AT CLOSEST APPROACH OF THE TARGET PLANET.  IF ITARG
C                    =3,4  THIS IS THE TIME AT SPHERE OF INFLUENCE OF
C                    THE TARGET PLANET.
C        NBOD     -- THE NUMBER OF GRAVITATIONAL BODIES TO BE CONSIDERED
C                    IN THE INTEGRATION.
C        NB       -- A VECTOR OF DIMENSION NBOD SPECIFYING THE INDICES
C                    OF THE GRAVITATIONAL BODIES.  THE SECOND BODY IS
C                    ASSUMED TO BE THE LAUNCH PLANET, THE THIRD, THE
C
C
C                    TARGET PLANET.  THE NUMBERING SYSTEM ASSIGNS THE
C                    INDEX 1 TO THE SUN, 2 TO MERCURY, 3 TO VENUS, 4 TO
C                    EARTH, 5 TO MARS, 6 TO JUPITER, 7 TO SATURN, 8 TO
C                    URANUS, 9 TO NEPTUNE, 10 TO PLUTO, AND 11 TO THE
C                    EARTHS MOON.
C        INJEK    -- A FLAG DESIGNATING WHICH OF TWO INJECTION OPTIONS
C                    IS TO BE USED.
C                         IF INJEK = 1 - THE POINT-TO-POINT CONDITIONS
C                                        ARE TO BE COMPUTED AND USED AS
C                                        THE ZERO ITERATE INJECTION
C                                        CONDITIONS.
C                                  = 2 - THE ZERO ITERATE INJECTION
C                                        CONDITIONS ARE READ IN.
C        RS       -- THE ZERO ITERATE INJECTION POSITION AND VELOCITY IN
C                    HELIOCENTRIC ECLIPTIC COORDINATES.  IF INJEK = 1,
C                    THE CORRESPONDING COLUMNS ARE LEFT BLANK.
C        ITARG    -- A FLAG DESIGNATING WHICH OF SIX TARGET OPTIONS ARE
C                    TO BE IN EFFECT.  THE OPTIONS ARE
C                      ITARG  OPTION
C                        1    POINT-TO-POINT CONDITIONS
C                        2    PATCHED CONIC CONDITIONS (UNBIASED PTP)
C                        3    B.T, B.R, APPROXIMATE TSI
C                        4    B.T, B.R, TSI
```

```
C                              5     APPROXIMATE RCA, ICA, TCA
C                              6     EXACT RCA, ICA, TCA
C         TARG1,   -    TARGET PARAMETERS.  THE PARAMETERS HAVE THE FOLLOW-
C          TARG2        ING DEFINITIONS DEPENDING ON THE TARGET OPTION.
C                         1TARG        TARG1       TARG2
C                          1,2            DO NOT APPLY
C                          3,4       B.T (KM)    B.R (KM)
C                          5,6       INC (DEG)   RCA (KM)
C         TOL1,   --   TARGET TOLERANCES. THE TOLERANCES SPECIFY THE ERROR
C          TOL2,       THAT WILL BE ACCEPTABLE IN THE TARGET PARAMETERS AC-
C           TOL3       CORDING TO THE FOLLOWING SCHEME
C                        1TARG         TOL1        TOL2        TOL3
C                         1,2                    DO NOT APPLY
C                        3,4,5       B.T (KM)    B.R (KM)   TSI (DAYS)
C                          6         INC (DEG)   RCA (KM)   TCA (DAYS)
C         ISKEJ    --   A FLAG DESIGNATING THE NUMBER OF ACCURACY LEVELS TO
C                       BE USED IN THE TARGETING PROCESS.
C         AC       --   A VECTOR OF DIMENSION ISKEJ WHOSE COMPONENTS ARE
C                       THE PROGRESSIVE ACCURACY LEVELS FROM THE LOWEST TO
C                       THE DESIRED FINAL LEVEL.
C         NITS     --   THE MAXIMUM NUMBER OF ITERATIONS ALLOWED AT THE
C                       FINAL ACCURACY LEVEL.
C         INCPR    --   THE NUMBER OF INTEGRATION INCREMENTS BETWEEN EACH
C                       PRINTOUT OF TRAJECTORY INFORMATION IN THE FINAL
C                       INTEGRATION OF THE TARGETED INJECTION CONDITIONS.
C         TIMPR    --   THE NUMBER OF DAYS BETWEEN EACH PRINTOUT OF TRAJEC-
C                       TORY INFORMATION IN THE FINAL INTEGRATION OF THE
C                       TARGETED INJECTION CONDITIONS.
C         BDELV    --   THE BASIC VELOCITY INCREMENT BY WHICH THE NOMINAL
C                       VELOCITIES ARE PERTURBED IN COMPUTING STATE TRANSI-
C                       TION MATRICES.  IN OUTER TARGETING THE VELOCITY
C                       INCREMENT IS 10 TIMES GREATER, IN CLOSEST APPROACH
C                       TARGETING IT IS 1/10 AS LARGE.
C
C
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,SB,SBDT,SBDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      DIMENSION IDAT1(5),IDAT2(5),IDAT3(5),RS(6),R1(3),V1(3),P(3),Q(3),
     1 WPT(3),ECOP(3,3),OPEQ(3,3),ECEQP(3,3),EQECP(3,3),RSF(6),RSS(6),
     2 SRS(6),TRG1(4),TRG2(4),TRG3(4),Z(3),PHI(2,2),PSI(3,3),RQ(6)
      DIMENSION VEL(4,3),AC(5)
      AU=149598500.
      AUDAY=1731.4641203704
      RAD=57.2957795
      AUS=149598500.
      ALNGTH=AU
      TM=86400.
    1 CONTINUE
      CALL CPWMS(TIMS)
      WRITE(6,980)
      WRITE(6,901)
      WRITE(6,902)
      WRITE(6,903)
      IEPHEM = 1
      IPRINT = 1
      TRTM=0.
      READ (5,991)IDAT1,S1,IDAT2,S2
```

```
      WRITE(6,941)IDAT1,S1,IDAT2,S2
      READ (5,992)NBOD,NB
      WRITE(6,942)NBOD,(NB(I),I=1,NBOD)
      READ (5,993)INJEK,RS
      WRITE(6,943)INJEK,RS
      READ (5,994)ITARG,TARG1,TARG2,TOL1,TOL2,TOL3
      WRITE(6,944)ITARG,TARG1,TARG2,TOL1,TOL2,TOL3
      READ (5,995)ISKEJ,AC
      WRITE(6,945)ISKEJ,(AC(I),I=1,ISKEJ)
      READ (5,996)NITS,INCPR,TIMPR,BDELV
      WRITE(6,946)NITS,INCPR,TIMPR,BDELV
      ACC=AC(ISKEJ)
      MIDI=1
      IF(ITARG-6)10,5,5
    5 XTOL1=TOL1
      XTOL2=TOL2
      XTOL3=TOL3
   10 CONTINUE
      NLP=NB(2)
      NTP=NB(3)
      TSPH=SPHERE(NTP)*AUS
      TMU=PMASS(NTP)*(AUS**3/86400.**2)
      CALL TIME(D1,IDAT1(1),IDAT1(2),IDAT1(3),IDAT1(4),IDAT1(5),S1,0)
      CALL TIME(D2,IDAT2(1),IDAT2(2),IDAT2(3),IDAT2(4),IDAT2(5),S2,0)
      IF(ITARG-5)7,4,4
    4 DINCL=TARG1
      DRCA=TARG2
      D3=D2
      CALL PECEQ(NTP,D3,ECEQP)
      DO 37 I=1,3
      DO 37 J=1,3
   37 EQECP(I,J)=ECEQP(J,I)
      DO 6 I=1,5
    6 IDAT3(I)=IDAT2(I)
      S3=S2
      GO TO 12
    7 DBDT=TARG1
      DBDR=TARG2
      TARG3=D2
      IF(NTP-5)8,9,11
    8 D3=D2+1.
      GO TO 12
    9 D3=D2+1.5
      GO TO 12
   11 D3=D2+60.
   12 CONTINUE
      DEND=D3+10.
      DELTM=DEND-D1
      IF(NITS)320,320,17
   17 CONTINUE
      ITIM=0
      JC3=1
      IF(ITARG-2)14,13,14
   13 JC3=0
   14 CONTINUE
      JINJT=0
      IF(INJEK-1)34,15,34
   15 IF(ITARG-5)16,18,18
   16 IF(ITIM)18,19,18
   18 JINJT=1
```

```
19 CONTINUE
   CALL NJEXN(JC3,JINJT,NLP,NTP,D1,D3,R1,V1,VSI)
   GO TO (1,1,20,20,20,20),ITARG
20 RSI(1)=-VSI(1)*1.05
   RSI(2)=-VSI(2)*.95
   RSI(3)=-VSI(3)*1.05
   RM=SQRT(RSI(1)*RSI(1)+RSI(2)*RSI(2)+RSI(3)*RSI(3))
   RSI(1)=TSPH*RSI(1)/RM
   RSI(3)=TSPH*RSI(3)/RM
   RSI(2)=TSPH*RSI(2)/RM
   VX=SQRT(VSI(1)*VSI(1)+VSI(2)*VSI(2)+VSI(3)*VSI(3))
   IF(ITIM)25,21,25
21 GO TO (1,1,23,23,25,25),ITARG
23 CALL CONIC(RSI,TSPH,VSI,VX,A,E,HI,HL,HW,TA,P,Q,TMU,PERI,HP,WPT)
   VHE=SQRT(VX*VX-2.*TMU/TSPH)
   CTSE = 1./E*(HP/TSPH-1.)
   STSE = SQRT(1.-CTSE**2)
   DENE = PERI/TSPH*(1.+E)
   SFE = SQRT(E**2-1.)*STSE/DENE
   FE = ALOG(SFE+SQRT(SFE**2+1.))
   TSICA = TMU/VHE**3*(E*SFE-FE)
   D3=D2+TSICA/86400.
   CALL TIME(D3 ,IDAT3(1),IDAT3(2),IDAT3(3),IDAT3(4),IDAT3(5),S3,1)
   ITIM=1
   GO TO 14
25 DEND=D3+10.
   DELTM=DEND-D1
   IF(ITARG-5)31,27,27
27 CONTINUE
   DINCL=DINCL/RAD
   CALL CASOI(RSI,VSI,TMU,EQECP,DINCL,DRCA,DB,DBDT,DBDR,TSICA,DECL)
   DINCL=DINCL*RAD
   D2=D3-TSICA
   CALL TIME(D2 ,IDAT2(1),IDAT2(2),IDAT2(3),IDAT2(4),IDAT2(5),S2,1)
31 IF(INJEK-1)34,32,34
32 CONTINUE
   CALL TIME(D1 ,IDAT1(1),IDAT1(2),IDAT1(3),IDAT1(4),IDAT1(5),S1,1)
   CALL ORB(NLP,D1 )
   NO(1)=NLP
   CALL EPHEM(1,D1 ,1)
   RS(1)=R1(1)+XP(1)*AU
   RS(2)=R1(2)+XP(2)*AU
   RS(3)=R1(3)+XP(3)*AU
   RS(4)=V1(1)+XP(4)*AUDAY
   RS(5)=V1(2)+XP(5)*AUDAY
   RS(6)=V1(3)+XP(6)*AUDAY
34 CALL OUT1(ITARG,INJEK,NITS,NB,IDAT1,S1,IDAT2,S2,IDAT3,S3,DBDT,DBDR
  1 ,DINCL,DRCA,TOL1,TOL2,TOL3,ACC,RS,INCPR,TIMPR,NBOD,ISKEJ,AC,MIDI)
   WRITE(6,982)
   WRITE(6,950)
   WRITE(6,951)
   WRITE(6,931)
   IF(NITS)35,120,35
35 CONTINUE
   IPSI2=0
   ILS6=0
   INPR=10000
   DELTP=10000.
   NOSOI=0
36 LEVEL=0
```

```
      ISTM=0
      ICL2=0
      ISP2=1
   39 CONTINUE
      LEVEL=LEVEL+1
   40 CONTINUE
      PRERR=1.E+20
      ITER=0
C
C     SET PARAMETERS FOR SPECIFIC ACCURACY LEVEL
      ISTM=0
      IF(LEVEL-1)42,42,41
   41 ISTM=1
   42 CONTINUE
      IF(ISKEJ-LEVEL)105,43,44
   43 CONTINUE
      ACK=AC(ISKEJ)
      NITRS=NITS
      TOLR1=TOL1
      TOLR2=TOL2
      TOLR3=TOL3
      GO TO 48
   44 CONTINUE
      ACK=AC(LEVEL)
      NITRS=MIDI
      TOLR1=250.
      TOLR2=250.
      TOLR3=.02
      IF(LEVEL-1)105,45,48
   45 CONTINUE
      NITRS=8
   48 CONTINUE
      IF(LEVEL-2)50,49,50
   49 WRITE(6,932)
C     PREPARATIONS COMPLETED FOR SPECIFIC ACCURACY LEVEL
C
   50 CONTINUE
      ISTEP=4
   51 IF(ISTEP-4)52,53,53
   52 RS(ISTEP+3)=RS(ISTEP+3)+DELV
   53 ISPH=0
      ICL=0
      TIMINT=0
      INCMT=0
      DO 54 I=1,3
   54 VEL(ISTEP,I)=RS(I+3)
      CALL VMP(RS,ACK,D1,TRTM,DELTM,RSF,ISP2)
      IF(NITRS)125,125,56
   56 IF(ISPH)55,561,55
  561 IF(NOSOI)562,562,661
  562 IF(ITER)110,110,661
   55 IF(ICL2)63,57,63
   57 TRG1(ISTEP)=SBDT
      TRG2(ISTEP)=SBDR
      TRG3(ISTEP)=DSI
      IF(ITARG-5)65,59,59
   59 IF(ISTEP-4)65,60,65
   60 CONTINUE
      DINCL=DINCL/RAD
      CALL CASOI(RSI,VSI,TMU,EQECP,DINCL,DRCA,DB,DBDT,DBDR,TSICA,DECL)
```

```
      DINCL=DINCL*RAD
      D2=D3-TSICA
      CALL TIME(D2 ,IDAT2(1),IDAT2(2),IDAT2(3),IDAT2(4),IDAT2(5),S2,1)
      TARG1=DBDT
      TARG2=DBDR
      TARG3=D2
      GO TO 65
   63 CONTINUE
      TIMCR=-(RC(1)*RC(4)+RC(2)*RC(5)+RC(3)*RC(6))/(RC(4)*RC(4)+RC(5)*RC
     $(5)+RC(6)*RC(6))
      RM=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      VM=SQRT(RC(4)*RC(4)+RC(5)*RC(5)+RC(6)*RC(6))
      RD=(RC(1)*RC(4)+RC(2)*RC(5)+RC(3)*RC(6))/RM
      TIMCR=-RD*RM/(VM*VM-TMU/RM-RD*RD)
      RC(1)=RC(1)+RC(4)*TIMCR
      RC(2)=RC(2)+RC(5)*TIMCR
      RC(3)=RC(3)+RC(6)*TIMCR
      DC=DC+TIMCR/86400.
      RCA=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      DO 64 I=1,3
      RQ(I)=ECEQP(I,1)*RC(1)+ECEQP(I,2)*RC(2)+ECEQP(I,3)*RC(3)
   64 RQ(I+3)=ECEQP(I,1)*RC(4)+ECEQP(I,2)*RC(5)+ECEQP(I,3)*RC(6)
      Z(1)=RQ(2)*RQ(6)-RQ(3)*RQ(5)
      Z(2)=RQ(3)*RQ(4)-RQ(1)*RQ(6)
      Z(3)=RQ(1)*RQ(5)-RQ(2)*RQ(4)
      ZM=SQRT(Z(1)*Z(1)+Z(2)*Z(2)+Z(3)*Z(3))
      Z(1)=Z(1)/ZM
      Z(2)=Z(2)/ZM
      Z(3)=Z(3)/ZM
      AINCL=ACOS(Z(3))
      TRG1(ISTEP)=AINCL*RAD
      TRG2(ISTEP)=RCA
      TRG3(ISTEP)=DC
   65 CONTINUE
      ERR1=TRG1(ISTEP)-TARG1
      ERR2=TRG2(ISTEP)-TARG2
      ERR3=TRG3(ISTEP)-TARG3
      CALL CPWMS(TIMC)
      TIMC=TIMC-TIMS
      ISTP=0
      IF(ISTEP-4)66,67,67
   67 CONTINUE
      WRITE(6,973)
      WRITE(6,952)LEVEL,ITER,ISTP,ACK,RS(4),RS(5),RS(6),TRG1(4),TRG2(4),
     1 TRG3(4),TARG1,TARG2,TARG3,TIMINT,TIMC,INCMT
      IF(ICL2)676,670,676
  670 IF(NOSOI)676,671,676
  671 IF(ABS(ERR1)-100.)672,672,675
  672 IF(ABS(ERR2)-100.)673,673,675
  673 IF(ABS(ERR3)-.01)674,674,675
  674 DELV=BDELV/2.
      GO TO 676
  675 DELV=BDELV
  676 CONTINUE
C
C     BEGIN CHECK LOOP FOR BAD STEP
      FAC1=1.
      FAC2=1.
      FAC3=100000.
      IF(IPSI2)682,680,682
```

```
  680 CONTINUE
      IF(ITARG-3)679,681,679
  681 FAC3=0.
      GO TO 679
  682 CONTINUE
      FAC1=100.
  679 CONTINUE
      ERROR=FAC1*ABS(ERR1)+FAC2*ABS(ERR2)+FAC3*ABS(ERR3)
      WRITE(6,997)ERROR,PRERR
  997 FORMAT(1X,6HERROR=,E14.7,5X,6HPRERR=,E14.7)
      IF(ERROR-PRERR)662,662,661
  661 CONTINUE
      RED=.25
      DV1=DV1*RED
      DV2=DV2*RED
      DV3=DV3*RED
      RS(4)=SV1+DV1
      RS(5)=SV2+DV2
      RS(6)=SV3+DV3
      WRITE(6,906)RED
      ISTEP=4
      GO TO 51
  662 PRERR=ERROR
      SV1=RS(4)
      SV2=RS(5)
      SV3=RS(6)
C     END CHECK LOOP FOR BAD STEP
C
   66 ISTP=ISTEP
      IF(ABS(ERR1)-TOLR1)71,71,75
   71 IF(ABS(ERR2)-TOLR2)72,72,75
   72 IF(ITARG-3)73,100,73
   73 IF(ABS(ERR3)-TOLR3)100,100,75
   75 IF(ISTEP-3)77,77,83
   77 RS(ISTEP+3)=RS(ISTEP+3)-DELV
      IF(ISTEP-3)78,81,81
   78 IF(ITARG-3)79,76,79
   76 IF(ISTEP-2)79,80,80
   79 ISTEP=ISTEP+1
      GO TO 51
   80 PHI(1,1)=(TRG1(1)-TRG1(4))/DELV
      PHI(1,2)=(TRG1(2)-TRG1(4))/DELV
      PHI(2,1)=(TRG2(1)-TRG2(4))/DELV
      PHI(2,2)=(TRG2(2)-TRG2(4))/DELV
      CALL MATIN(PHI,PHI,2)
      WRITE(6,973)
      DO 801 I=1,2
  801 WRITE(6,954)LEVEL,ITER,I,ACK,VEL(I,1),VEL(I,2),VEL(I,3),TRG1(I),
     1 TRG2(I),TRG3(I),PHI(I,1),PHI(I,2)
      GO TO 85
   81 DO 82 I=1,3
      PSI(1,I)=(TRG1(I)-TRG1(4))/DELV
      PSI(2,I)=(TRG2(I)-TRG2(4))/DELV
   82 PSI(3,I)=(TRG3(I)-TRG3(4))/DELV
      CALL MATIN(PSI,PSI,3)
      WRITE(6,973)
      DO 821 I=1,3
  821 WRITE(6,953)LEVEL,ITER,I,ACK,VEL(I,1),VEL(I,2),VEL(I,3),TRG1(I),
     1 TRG2(I),TRG3(I),PSI(I,1),PSI(I,2),PSI(I,3)
      GO TO 85
```

```
 83 IF(ISTM)85,84,85
 84 ISTEP=1
    GO TO 51
 85 DT1=TARG1-TRG1(4)
    DT2=TARG2-TRG2(4)
    IF(ITARG-3)87,86,87
 86 DV1=PHI(1,1)*DT1+PHI(1,2)*DT2
    DV2=PHI(2,1)*DT1+PHI(2,2)*DT2
    DV3=0.
    GO TO 88
 87 DT3=TARG3-TRG3(4)
    DV1=PSI(1,1)*DT1+PSI(1,2)*DT2+PSI(1,3)*DT3
    DV2=PSI(2,1)*DT1+PSI(2,2)*DT2+PSI(2,3)*DT3
    DV3=PSI(3,1)*DT1+PSI(3,2)*DT2+PSI(3,3)*DT3
    RS(6)=RS(6)+DV3
 88 RS(4)=RS(4)+DV1
    RS(5)=RS(5)+DV2
    ITER=ITER+1
    WRITE(6,973)
    IF(ITER-NITRS)89,100,100
 89 ISTEP=4
    GO TO 51
100 LEVEL=LEVEL+1
    IF(ITARG-6)40,101,101
101 IF(LEVEL-2)40,102,40
102 DO 103 I=1,6
103 RSS(I)=RS(I)
    GO TO 40
105 IF(NOSOI)111,106,111
106 IF(ITARG-6)120,107,120
107 IF(IPSI2)115,109,115
109 IPSI2=1
    ICL2=1
    DELV=.1*BDELV
    DO 112 I=1,6
    SRS(I)=RS(I)
112 RS(I)=RSS(I)
    DECL=DECL*RAD
    SM=1.
    IF(ABS(DINCL)-90.)210,210,201
201 SM=-1.
    DINCL=180.-ABS(DINCL)
210 CONTINUE
    DINCL=ABS(DINCL)
    IF(DINCL-ABS(DECL))215,225,225
215 DINCL=ABS(DECL)
    IF(SM)220,225,225
220 DINCL=180.-DINCL
225 CONTINUE
    TOLR1=0.
    TOLR2=0.
    TOLR3=0.
    TARG1=DINCL
    TARG2=DRCA
    TARG3=D3
    ISP2=0
    ACK=AC(1)
    NITRS=3
    NITRS=1
    ISTM=0
```

```
      ISKEJ=1
      LEVEL=1
      WRITE(6,933)
      GO TO 50
  115 IF(ILS6)120,116,120
  116 ILS6=1
      DO 117 I=1,6
  117 RS(I)=SRS(I)
      TOLR1=XTOL1
      TOLR2=XTOL2
      TOLR3=XTOL3
      ISKEJ=1
      LEVEL=1
      ACK=ACC
      NITRS=NITS
      ISTM=1
      WRITE(6,934)
      GO TO 50
  120 CONTINUE
      WRITE(6,980)
      WRITE(6,920)
      WRITE(6,921)
      WRITE(6,922)IDAT1,S1
      WRITE(6,923)
      RSM=SQRT(RS(1)*RS(1)+RS(2)*RS(2)+RS(3)*RS(3))
      VSM=SQRT(RS(4)*RS(4)+RS(5)*RS(5)+RS(6)*RS(6))
      WRITE(6,924)(RS(I),I=1,3),RSM
      WRITE(6,925)(RS(I),I=4,6),VSM
      WRITE(6,926)
      CALL ORB(NLP,D1)
      NO(1)=NLP
      CALL EPHEM(1,D1,1)
      DO 330 I=1,3
      RC(I)=RS(I)-XP(I)*AU
  330 RC(I+3)=RS(I+3)-XP(I+3)*AUDAY
      RSM=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      VSM=SQRT(RC(4)*RC(4)+RC(5)*RC(5)+RC(6)*RC(6))
      WRITE(6,924)(RC(I),I=1,3),RSM
      WRITE(6,925)(RC(I),I=4,6),VSM
      WRITE(6,927)
      WRITE(6,928)
      IF(ITARG-5)300,300,310
  300 WRITE(6,929)TARG1,TARG2,IDAT2,S2
      WRITE(6,912)
      CALL TIME(DSI,IDAT2(1),IDAT2(2),IDAT2(3),IDAT2(4),IDAT2(5),S2,1)
      WRITE(6,929)TRG1(ISTEP),TRG2(ISTEP),IDAT2,S2
      GO TO 320
  310 CONTINUE
      CALL TIME(DC,IDAT3(1),IDAT3(2),IDAT3(3),IDAT3(4),IDAT3(5),S3,1)
      WRITE(6,911)TARG1,TARG2,IDAT3,S3
      WRITE(6,912)
      WRITE(6,911)TRG1(ISTEP),TRG2(ISTEP),IDAT3,S3
  320 CONTINUE
      ISP2=0
      NITRS=0
      ACK=ACC
      ICL2=1
      INPR=INCPR
      DELTP=TIMPR
      IPRINT=0
```

```
      GO TO 53
  110 NOSOI=1
      SSPH=SPHERE(NTP)
      ITARGS=ITARG
      STARG1=TARG1
      STARG2=TARG2
      STARG3=TARG3
      R=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      SPHERE(NTP)=1.2*R/AUS
      TARG1=0.
      TARG2=0.
      V=SQRT(RC(4)*RC(4)+RC(5)*RC(5)+RC(6)*RC(6))
      TMDF=(SPHERE(NTP)*AUS-TSPH)/(V*86400.)
      TARG3=D2-TMDF
      TOLR1=100000.
      TOLR2=100000.
      TOLR3=1.
      ISSKJ=ISKEJ
      ISKEJ=1
      SDELV=DELV
      DELV=10.*BDELV
      DELV=50.*BDELV
      LEVEL=1
      NITRS=5
      NITRS=8
      ITARG=4
      V=SPHERE(NTP)*AUS
      WRITE(6,930)R,DC,V
      SV1=RS(4)
      SV2=RS(5)
      SV3=RS(6)
      GO TO 50
  111 SPHERE(NTP)=SSPH
      ITARG=ITARGS
      TARG1=STARG1
      TARG2=STARG2
      TARG3=STARG3
      TOLR1=TOL1
      TOLR2=TOL2
      TOLR3=TOL3
      ISKEJ=ISSKJ
      DELV=SDELV
      NOSOI=0
      WRITE(6,990)
      GO TO 36
  125 CONTINUE
      GO TO 1
  901 FORMAT(/39X,50HI N T E R P L A N E T A R Y    T R A J E C T O R Y)
  902 FORMAT(/47X,34HT A R G E T I N G    P R O G R A M)
  903 FORMAT(///10X,10HINPUT DATA)
  906 FORMAT( 1X,50HBAD STEP ..... CORRECTION IS REDUCED BY FACTOR OF ,
     $F4.3)
  911 FORMAT(20X,12HINCLINATION=,F9.3,5X,6HRCA=   ,E14.7,5X,4HTCA=,5I5,F9
     $.3)
  912 FORMAT(15X,28HINTEGRATED TRAJECTORY VALUES)
  920 FORMAT(/////10X,17HTARGETING SUMMARY)
  921 FORMAT(//10X,20HINJECTION CONDITIONS)
  922 FORMAT(15X,14HCALENDAR DATE=,5I5,F9.3)
  923 FORMAT(15X,33HHELIOCENTRIC ECLIPTIC COORDINATES)
  924 FORMAT(20X,9HPOSITION=,3(2X,E18.11),2X,10HMAGNITUDE=,2X,E18.11)
```

```
925 FORMAT(20X,9HVELOCITY=,3(2X,E18.11),2X,10HMAGNITUDE=,2X,E18.11)
926 FORMAT(15X,35HPLANETOCENTRIC ECLIPTIC COORDINATES)
927 FORMAT(//10X,17HTARGET CONDITIONS)
928 FORMAT(15X,17HTARGET PARAMETERS)
929 FORMAT(20X,4HB.T=,F11.2,5X,4HB.R=,F11.2,5X,4HTSI=,5I5,F9.3)
930 FORMAT(/1X,39HOUTER TARGETING ..... CLOSEST APPROACH=,E14.7,12H ..
   $... DATE=,F10.3,45H ..... ARTIFICIAL SPHERE-OF-INFLUENCE RADIUS=,
   $ E14.7)
931 FORMAT(//1X,73HTARGETING AND CONSTRUCTION OF SPHERE-OF-INFLUENCE S
   $TATE TRANSITION MATRIX)
932 FORMAT(/1X, 43HTARGETING TO SPHERE-OF-INFLUENCE CONDITIONS)
933 FORMAT(/1X, 56HCONSTRUCTION OF CLOSEST-APPROACH STATE TRANSITION M
   $ATRIX)
934 FORMAT(/1X,40HTARGETING TO CLOSEST-APPROACH CONDITIONS)
941 FORMAT( 20X,15HINJECTION DATE=,I5,4I3,F7.3, 9X,12HTARGET DATE=,I5,
   $4I3,F7.3)
942 FORMAT( 20X,5HNBOD=,I3,4X,7HBODIES=,11I3)
943 FORMAT( 20X,6HINJEK=,I2,4X,6HSTATE=,1X,3(F15.3,1X),3(F11.6,1X))
944 FORMAT( 20X,6HITARG=,I2,4X,8HTARGETS=,2F10.2,8X,11HTOLERANCES=,2F1
   $0.2,3X,F5.3)
945 FORMAT( 20X,6HISKEJ=,I2,4X,16HACCURACY LEVELS=,6E11.2)
946 FORMAT( 20X,5HNITS=,I3,4X,6HINCPR=,I6,5X,6HTIMPR=,F8.3,5X,6HBDELV=
   $,E9.2)
950 FORMAT(//10H L I S   A,     9X,1HX,10X,1HY,10X,1HZ,7X,80HTRAJECTOR
   1Y TRAJECTORY TRAJECTORY    TARGET     TARGET     TARGET  TIME TOTAL
   2   NO,/1X,10HV T T    C,9X,1HD,10X,1HD,10X,1HD,8X,3HB.T,8X,3HB.R,8
   3X,3HTSI,7X,46HB.T/INCL   B.R/RCA    TSI/TCA    PER    CP     OF)
951 FORMAT(1X,11HE E E     C,9X,1HO,2(10X,1HO),3(9X,2HOR),36X,17HINTEG
   2  TIME INTEG,/1X,12HL R P       Y,9X,1HT,2(10X,1HT),9X,4HINCL,7X,3H
   2RCA,8X,3HTCA,7X,44HSTATE TRANSITION MATRIX    (SEC) (SEC)  INCR)
952 FORMAT(3I2,E9.2,3F11.6,2F11.2,F10.3,2F11.2,F10.3,F6.2,F6.1,I6)
953 FORMAT(3I2,E9.2,3F11.6,2F11.2,F10.3,2X,3(E9.2,1X))
954 FORMAT(3I2,E9.2,3F11.6,2F11.2,F10.3,6X,E9.2,2X,E9.2)
973 FORMAT(1H )
980 FORMAT(1H1)
982 FORMAT(1X,32HNUMERICAL DIFFERENCING PROCEDURE)
990 FORMAT(/1X,15HINNER TARGETING)
991 FORMAT(I5,4I3,F7.3,5X,I4,4I3,F7.3)
992 FORMAT(I2,3X,11I5)
993 FORMAT(I2,2X,3E15.8,3E10.3)
994 FORMAT(I2,2X,5F15.5)
995 FORMAT(I2,6X,7F10.8)
996 FORMAT(I2,7X,I5,5X,F9.4,F11.8)
    STOP
    END
```

```
      SUBROUTINE ACTB(R,V,GMX,B,BDT,BDR)
      DIMENSION R(3),V(3),WV(3),Z(3),PV(3),QV(3),SV(3),BV(3),TV(3),RV(3)
      WV(1) = R(2)*V(3) - R(3)*V(2)
      WV(2) = R(3)*V(1) - R(1)*V(3)
      WV(3) = R(1)*V(2) - R(2)*V(1)
      C1 = SQRT(WV(1)**2 + WV(2)**2 + WV(3)**2)
      WV(1) = WV(1)/C1
      WV(2) = WV(2)/C1
      WV(3) = WV(3)/C1
      RRD = R(1)*V(1) + R(2)*V(2) + R(3)*V(3)
      RM=SQRT(R(1)**2+R(2)**2+R(3)**2)
      VM=SQRT(V(1)**2+V(2)**2+V(3)**2)
      RD=RRD/RM
      P=C1**2/GMX
      A=RM/(2.-RM*VM**2/GMX)
      E=SQRT(1.-P/A)
      CTA=(P-RM)/(E*RM)
      STA=RD*C1/(E*GMX)
      B=SQRT(P*ABS(A))
      AB=SQRT(A**2+B**2)
      DO 10 I=1,3
      Z(I)= RM/C1*V(I)-RD/C1*R(I)
      PV(I)= CTA*R(I)/RM-STA*Z(I)
      QV(I)=STA*R(I)/RM+CTA*Z(I)
      SV(I)=-A/AB*PV(I)+B/AB*QV(I)
   10 BV(I)=B**2/AB*PV(I)+A*B/AB*QV(I)
      AB=SQRT(SV(1)**2+SV(2)**2)
      TV(1)=SV(2)/AB
      TV(2)=-SV(1)/AB
      TV(3)=0.
      RV(1) = SV(2)*TV(3) - SV(3)*TV(2)
      RV(2) = SV(3)*TV(1) - SV(1)*TV(3)
      RV(3) = SV(1)*TV(2) - SV(2)*TV(1)
      AB = SQRT(RV(1)**2 + RV(2)**2 + RV(3)**2)
      RV(1) = RV(1)/AB
      RV(2) = RV(2)/AB
      RV(3) = RV(3)/AB
      BDT = BV(1)*TV(1) + BV(2)*TV(2) + BV(3)*TV(3)
      BDR = BV(1)*RV(1) + BV(2)*RV(2) + BV(3)*RV(3)
      RETURN
      END
```

```
      SUBROUTINE AUX(W,ELAT,ELON,AZ,PV,Q,TAI,ANG1,ANG2,TIM1,TIM2,S,E,
     1               RP,GME,ROT,DJL,TL,TB,PHI,THI,RAI,AZI,TINJ,TC)
      DIMENSION W(3),RL(3),PV(3),Q(3),RI(3),S(3)
       RAD=57.2957795
       DGTR=.0174532924
       PI=3.1415926536
      TAR=TAI/RAD
      STAI=SIN(TAR)
      CTAI=COS(TAR)
      SEL=SIN(ELAT/RAD)
      CEL=COS(ELAT/RAD)
      SAZ=SIN(AZ/RAD)
      CAZ=COS(AZ/RAD)
      WZ=W(3)**2-1.
      CRA=(W(1)*SAZ*SEL+W(2)*CAZ)/WZ
      SRA=(W(2)*SAZ*SEL-W(1)*CAZ)/WZ
      IF(CRA)10,11,10
   10 RALS = ATAN(SRA/CRA)
      IF(CRA)12,11,13
   11 RALS = PI/2.
      IF(SRA)12,13,13
   12 RALS = RALS + PI
   13 IF(RALS)14,15,15
   14 RALS = 2. *PI + RALS
   15 RL(1) = CRA*CEL
      RL(2)=SRA*CEL
      RL(3)=SEL
      CRA = RL(1) * PV(1) + RL(2) * PV(2) + RL(3) * PV(3)
      SRA = RL(1)*Q(1) + RL(2)*Q(2) + RL(3)*Q(3)
      IF(CRA) 40,41,40
   40 TAL = ATAN(SRA/CRA)
      IF(CRA) 42,41,43
   41 TAL = PI/2.
      IF(SRA)42,43,43
   42 TAL = TAL + PI
   43 IF(TAL)44,45,45
   44 TAL = 2. * PI + TAL
   45 ALI = 2. * PI - TAL + TAR
      TC=ALI-(ANG1+ANG2)/RAD
      TC=TC*SQRT(RP**3/GME)
      TB=TC+TIM1+TIM2
      DO 21 I=1,3
   21 RI(I)=PV(I)*CTAI+Q(I)*STAI
      IF(RI(1)) 70,71,70
   70 RAI = ATAN(RI(2)/RI(1))
      IF(RI(1))72,71,73
   71 RAI = PI / 2.
      IF(RI(2)) 72,73,73
   72 RAI = RAI + PI
   73 IF(RAI) 74,75,75
   74 RAI = 2. * PI + RAI
   75 RAI = RAI*RAD
      PHI = ATAN(RI(3)/SQRT(RI(1)**2 + RI(2)**2))*RAD
      THI=ELON +RAI-RALS*RAD-ROT*TB/3600.
      THI=THI*DGTR
      R = THI/6.28318531
      N = R
      X = N
      THI = (R-X)*6.28318531
      CTAM=-1./E
```

```
      STAM=SQRT(1.-CTAM**2)
      IF(CTAM) 50,51,50
   50 TAM = ATAN(STAM/CTAM)
      IF(CTAM)52,51,53
   51 TAM = PI/2.
      IF(STAM)52,53,53
   52 TAM = TAM + PI
   53 IF(TAM ) 54,55,55
   54 TAM = 2. * PI + TAM
   55 CAZ = S(3)-COS(TAM-TAR)*SIN(PHI/RAD)
      CAZ=CAZ/(SIN(TAM-TAR)*COS(PHI/RAD))
      SAZ=SQRT(1.-CAZ**2)
      IF(CAZ)60,61,60
   60 AZI = ATAN(SAZ/CAZ)
      IF(CAZ)62,61,63
   61 AZI = PI / 2.
      IF(SAZ)62,63,63
   62 AZI = AZI + PI
   63 IF(AZI)64,65,65
   64 AZI = 2.*PI+ AZI
   65 AZI = AZI * RAD
      THI=THI*RAD
      D50=DJL-18262.5
      GHA=100.07554260+0.9856473460*D50+2.9015E-13*D50**2
      GHA=GHA*DGTR
      R = GHA/6.28318531
      N = R
      X = N
      GHA = (R-X)*6.28318531
      TL=RALS-ELON *DGTR-GHA
      R = TL/6.28318531
      N = R
      X = N
      TL = (R-X)*6.28318531
      TL=TL*RAD/ROT
      IF(TL)35,36,36
   35 TL=TL+24.
   36 TINJ=TL+TB/3600.
      RETURN
      END
```

```
      SUBROUTINE BIAS(MCODE,BVAL)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      DIMENSION BVAL(4)
      IF(MCODE-9) 10,30,40
10    IF(MCODE/2*2.EQ.MCODE) GO TO 20
      BVAL(1)=BIA(MCODE+1)
      GO TO 50
20    BVAL(1)=BIA(MCODE-1)
      BVAL(2)=BIA(MCODE)
      GO TO 50
30    BVAL(1)=BIA(9)
      BVAL(2)=BIA(10)
      BVAL(3)=BIA(11)
      GO TO 50
40    BVAL(1)=BIA(12)
50    RETURN
      END
```

```
      BLOCK DATA
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DATA CN/ 0.122223328,0.324776685E-04,-0.319770295E-06,0.0,
     A0.8228518595,0.2068578774E-01,0.3034933644E-05,0.0,1.3246996178,
     B0.2714840259E-01,0.5143873156E-05,0.0,0.20561421,0.2046E-04,
     C-0.3E-07,0.0,1.785111955,0.7142471E03,0.872664626E-08,0.0,
     D0.0592300268,0.1755510339E-04,-0.1696847884E-07,0.0,1.32260435,
     E0.1570534527E-01,0.7155849933E-05,0.0,2.2717874591,0.2457486613E-1
     F,0.1704120089E-04,0.0,0.682069E-02,-0.4774E-04,0.91E-07,0.0,
     G3.710626172,0.2796244623E03,1.682497399E-06,0.0,8*0.0,1.7666368138
     H,0.3000526417E-01,0.7902463002E-05,0.5817764173E-07,0.01675104,
     I-0.418E-04,-0.126E-06,0.0,6.256583781,0.1720196977E03,-0.195476876
     J2E-06,-0.1221730476E-8,0.0322944089,-0.1178097245E-04,0.2201054112
     KE-06,0.0,0.8514840375,0.1345634309E-01,-0.2424068406E-07,-0.930842
     L2677E-07,5.8332085089,0.3212729365E-01,0.2266503959E-05,
     M-0.2084698829E-07,0.09331290,0.92064E-04,-0.77E-07,0.0,5.576840523
     N,0.9145887726E+02,0.2365444735E-06,0.436332313E-09 /
      DATA ST/ 0.022841027,-0.9696273622E-04,     1.735518077,
     A0.1764479392E-01,     0.2218561704,0.2812302353E-01,     0.0483376,
     B0.16302E-03,     3.93135411,0.1450191928E02,     0.0435037861,
     C-0.7757018898E-07,     1.9684445802,0.152397787E-01,
     D1.5897996653,0.3419861162E-01,     0.05589,-0.34705E-03,
     E3.042621043,0.5837120844E01,     0.013486547,0.9696273622E-05,
     F1.2826407705,0.8912087493E-02,     2.9502426085,0.2834608631E-01,
     G     0.0470463,0.27204E-03,     1.2843599198,0.204654884E01,
     H0.0310537707,-0.1599885148E-03,     2.2810642235,0.1923032859E-01,
     I     0.7638202701,0.1532704516E-01,     0.852849E-02,0.7701E-04,
     J0.7204851506,0.1033089473E01,     0.2996712872,     0.0,1.91786587,
     K  0.0,3.91233424,  0.0,0.250236,  0.0,4.000815984,0.6944665094   /
      DATA SMJR/0.3870986,0.0,0.7233316,0.0,1.0000003,0.0,1.5236915,0.0,
     A5.202803,0.0,9.538843,0.0,19.182281,-0.57008E-3,30.057053,0.002101
     B66,39.43871,0,0/
      DATA EMN/4.523601515,-0.000924220,0.000036267,0.000000034,5.835151
     A540,0.001944367,-0.000180205,-0.000000209,4.719966573,0.229971481,
     B-0.000019774,0.000000033,0.099804108,0.054900489,.00256954448/
      DATA PMASS/2.959122083E-4,4.85E-11,7.243E-10,8.88757E-10,9.576E-11
     1,2.8252E-7,8.454E-8,1.290E-8,1.5E-8,7.4E-10,1.0921745E-11/
      DATA PI/3.141592653589793/
      DATA RAD/57.29577951303232/
      DATA LINPGE/60/
      DATA RMASS /
     11.0                 ,1.638999630283250E-7,2.447685427245681E-6,
     23.003448235900313E-6,3.236094940122147E-7,9.547426299951027E-4,
     32.856928427714348E-4,4.359401078485358E-5,5.069071021494582E-5,
     42.500741703937329E-6,3.695641978467207E-8                          /
      DATA RADIUS/
     A.00466582   ,.00001617   ,.00004044   ,.00004263   ,.00002279   ,
     B.00047727   ,.00040374   ,.00015761   ,.00014906   ,.00004679   ,
     C.00001161                                                        /
      DATA F/176*0.0/,ELMNT/80*0./
      DATA MONTH/10H   JANUARY,10H  FEBRUARY,10H     MARCH,10H     APRIL
     $,10H       MAY,10H      JUNE,10H      JULY,10H    AUGUST,
     $10H SEPTEMBER,10H   OCTOBER,10H  NOVEMBER,10H  DECEMBER/
      DATA PLANET/10HSUN       ,10HMERCURY   ,10HVENUS     ,10HEARTH
```

```
1 ,10HMARS      ,10HJUPITER   ,10HSATURN    ,10HURANUS    ,
210HNEPTUNE   ,10HPLUTO      ,10HMOON       /
 DATA SPHERE/0.,0.000746,0.00412,0.00618,0.00378,0.3216,0.3246,
$.346,0.5805,0.2366,0./
 END
```

```
      SUBROUTINE CASOI(RS,VHP,TTG,EQEC,DINCL,DRCA,DB,DBDT,DBDR,TSICA,
     $ DECL)
      DIMENSION VHP(3),S(3),EQEC(3,3),ECIM(3,3),EQIM(3,3),C(3),CIMP(3)
     1 ,VVP(3),RS(3),WPT(3),P(3),Q(3)
      DIMENSION Z(3)
C     THIS SUBROUTINE COMPUTES B.T AND B.R CORRESPONDING
C     TO SPECIFIED INCLINATION AND CLOSEST APPROACH.  VHP IS THE
C     VELOCITY VECTOR IN PLANETOCENTRIC ECLIPTIC COORDINATES.
C     TTG IS THE MU OF THE TARGET PLANET IN CORRESPONDING UNITS.
      PI=3.1415926536
      RAD=57.2957795
      SM = 1.
      SI = 1.
      SD = 1.
      SINCL=DINCL
      VHPM = SQRT(VHP(1)*VHP(1) + VHP(2)*VHP(2) + VHP(3)*VHP(3))
      VVP(1)=VHP(1)/VHPM
      VVP(2)=VHP(2)/VHPM
      VVP(3)=VHP(3)/VHPM
      TAND=VVP(3)/SQRT(1.-VVP(3)*VVP(3))
      DVP=ATAN(TAND)
      RVP = ATAN2(VVP(2),VVP(1))
      S(1)=EQEC(1,1)*VVP(1)+EQEC(2,1)*VVP(2)+EQEC(3,1)*VVP(3)
      S(2)=EQEC(1,2)*VVP(1)+EQEC(2,2)*VVP(2)+EQEC(3,2)*VVP(3)
      S(3)=EQEC(1,3)*VVP(1)+EQEC(2,3)*VVP(2)+EQEC(3,3)*VVP(3)
      TAND=S(3)/SQRT(1.-S(3)*S(3))
      DECL = ATAN(TAND)
      RA = ATAN2(S(2),S(1))
      IF(ABS(DINCL)-PI/2.)25,25,10
   10 SM = -1.
      IF(DINCL)20,20,15
   15 DINCL = DINCL - PI
      GO TO 25
   20 DINCL = PI + DINCL
   25 CONTINUE
      IF(DINCL)30,35,35
   30 SI = -1.
   35 IF(DECL)40,45,45
   40 SD = -1.
   45 CONTINUE
      IF(SI*DINCL-SD*DECL)47,48,48
   47 DINCL = SI*SD*DECL
      DINCL=DINCL*RAD
      WRITE(6,904)DINCL
      DINCL=DINCL/RAD
      W=RA-SI*PI/2.
      GO TO 49
   48 CONTINUE
      SDELW=SIN(DECL)*COS(DINCL)/(COS(DECL)*SIN(DINCL))
      TDELW=SDELW/SQRT(1.-SDELW*SDELW)
      W=RA-ATAN(TDELW)
   49 CONTINUE
      IF(SI)50,55,55
   50 W = W + PI
   55 IF(SM)60,65,65
   60 W = W + PI
   65 CONTINUE
      C(1) = SIN(W)*S(3)
      C(2) = -COS(W)*S(3)
      C(3) = COS(W)*S(2) - SIN(W)*S(1)
```

```
      CM=SQRT(C(1)*C(1)+C(2)*C(2)+C(3)*C(3))
      SCZ = 1.
      IF(C(3)) 70,75,75
   70 SCZ = -1.
   75 CONTINUE
      C(1) = SM*SCZ*C(1)/CM
      C(2) = SM*SCZ*C(2)/CM
      C(3) = SM*SCZ*C(3)/CM
      SND=SIN(DVP)
      CSD=COS(DVP)
      SNW=SIN(RVP)
      CSW=COS(RVP)
      ECIM(1,1)=SNW
      ECIM(1,2)=-CSW
      ECIM(1,3)=0.
      ECIM(2,1)=CSW*SND
      ECIM(2,2)=SNW*SND
      ECIM(2,3)=-CSD
      ECIM(3,1)=CSW*CSD
      ECIM(3,2)=SNW*CSD
      ECIM(3,3)=SND
      EQIM(1,1)=ECIM(1,1)*EQEC(1,1)+ECIM(1,2)*EQEC(2,1)+ECIM(1,3)*EQEC(3
     .,1)
      EQIM(1,2)=ECIM(1,1)*EQEC(1,2)+ECIM(1,2)*EQEC(2,2)+ECIM(1,3)*EQEC(3
     .,2)
      EQIM(1,3)=ECIM(1,1)*EQEC(1,3)+ECIM(1,2)*EQEC(2,3)+ECIM(1,3)*EQEC(3
     .,3)
      EQIM(2,1)=ECIM(2,1)*EQEC(1,1)+ECIM(2,2)*EQEC(2,1)+ECIM(2,3)*EQEC(3
     .,1)
      EQIM(2,2)=ECIM(2,1)*EQEC(1,2)+ECIM(2,2)*EQEC(2,2)+ECIM(2,3)*EQEC(3
     .,2)
      EQIM(2,3)=ECIM(2,1)*EQEC(1,3)+ECIM(2,2)*EQEC(2,3)+ECIM(2,3)*EQEC(3
     .,3)
      EQIM(3,1)=ECIM(3,1)*EQEC(1,1)+ECIM(3,2)*EQEC(2,1)+ECIM(3,3)*EQEC(3
     .,1)
      EQIM(3,2)=ECIM(3,1)*EQEC(1,2)+ECIM(3,2)*EQEC(2,2)+ECIM(3,3)*EQEC(3
     .,2)
      EQIM(3,3)=ECIM(3,1)*EQEC(1,3)+ECIM(3,2)*EQEC(2,3)+ECIM(3,3)*EQEC(3
     .,3)
      CIMP(1) = EQIM(1,1)*C(1) + EQIM(1,2)*C(2) + EQIM(1,3)*C(3)
      CIMP(2) = EQIM(2,1)*C(1) + EQIM(2,2)*C(2) + EQIM(2,3)*C(3)
      CIMP(3) = EQIM(3,1)*C(1) + EQIM(3,2)*C(2) + EQIM(3,3)*C(3)
      DB=DRCA*SQRT(1.+2.*TTG/(DRCA*VHPM*VHPM))
      THETA = ATAN2(CIMP(2),CIMP(1))
      THETA=THETA+PI/2.
      DBDT=DB*COS(THETA)
      DBDR=DB*SIN(THETA)
      RSM=SQRT(RS(1)**2+RS(2)**2+RS(3)**2)
      WPT(1) = RS(2)*VHP(3) - RS(3)*VHP(2)
      WPT(2) = RS(3)*VHP(1) - RS(1)*VHP(3)
      WPT(3) = RS(1)*VHP(2) - RS(2)*VHP(1)
      C1 = SQRT(WPT(1)**2 + WPT(2)**2 + WPT(3)**2)
      WPT(1) = 1. * WPT(1)/C1
      WPT(2) = 1. * WPT(2)/C1
      WPT(3) = 1. * WPT(3)/C1
      RRD = RS(1)*VHP(1) + RS(2)*VHP(2) + RS(3)*VHP(3)
      RD = RRD/RSM
      HP = C1**2/TTG
      A = RSM/(2.-RSM*VHPM**2/TTG)
      E = SQRT(1.-HP/A)
```

```
      PERI = HP/(1.+E)
      HI = ATAN(SQRT(1.-WPT(3)**2)/WPT(3))
      HL = ATAN2(WPT(1),-WPT(2))
      CTA = (HP-RSM)/(E*RSM)
      STA = RD*C1/(E*TTG)
      TA = ATAN2(STA,CTA)
      DO 11 I = 1,3
      Z(I) = RSM/C1*VHP(I)-RD/C1*RS(I)
      P(I) = CTA*RS(I)/RSM-STA*Z(I)
   11 Q(I) = STA*RS(I)/RSM+CTA*Z(I)
      HW = ATAN2(P(3),Q(3))
      VHE=SQRT(VHPM*VHPM-2.*TTG/RSM)
      CTS=1./E*(HP  /RSM-1.)
      STS = SQRT(1.-CTS**2)
      DEN = PERI/RSM*(1.+E)
      SF = SQRT(E**2-1.)*STS/DEN
      F = ALOG(SF+SQRT(SF**2+1.))
      TSICA = TTG/VHE**3*(E*SF-F)
      TSICA=TSICA/86400.
      DINCL=SINCL
  901 FORMAT(3(E20.13,5X))
  904 FORMAT(/1X,89HTHE DESIRED INCLINATION MUST BE SET EQUAL TO THE DEC
     $LINATION OF THE APPROACH ASYMPTOTE = ,F7.2)
      RETURN
      END
```

```
      SUBROUTINE CONC2(R,V,DELT,GMX,PSIEC)
C
C
C     COMPUTES THE 6 X 6 STATE TRANSITION MATRIX WHICH RELATES
C     PERTURBATIONS AT T1 TO PERTURBATIONS AT T2
C
C     INPUT IS LISTED BELOW
C        R       --  POSITION COMPONENTS OF THE VEHICLE RELATIVE TO THE
C                    GOVERNING BODY
C        V       --  VELOCITY COMPONENTS OF THE VEHICLE RELATIVE TO THE
C                    GOVERNING BODY
C        DELT    --  DELTA TIME (T2 - T1)
C        GMX     --  GRAVITATIONAL CONSTANT OF GOVERNING BODY
C
C     THE STATE TRANSITION MATRIX IS RETURNED AS PSIEC
C
C
      DIMENSION  V(3),R(3),WV(3),PV(3),Q(3),Z(3)
      DIMENSION  OPEC(6,6),FMI1(6,6),FM1(6,6),PSIOP(6,6),PSIEC(6,6)
      PI=3.1415926536
      RM=SQRT(R(1)*R(1)+R(2)*R(2)+R(3)*R(3))
      VM=SQRT(V(1)*V(1)+V(2)*V(2)+V(3)*V(3))
      WV(1)=R(2)*V(3)-R(3)*V(2)
      WV(2)=R(3)*V(1)-R(1)*V(3)
      WV(3)=R(1)*V(2)-R(2)*V(1)
      C1=SQRT(WV(1)*WV(1)+WV(2)*WV(2)+WV(3)*WV(3))
      WV(1)=WV(1)/C1
      WV(2)=WV(2)/C1
      WV(3)=WV(3)/C1
      RRD=R(1)*V(1)+R(2)*V(2)+R(3)*V(3)
      RD=RRD/RM
      P=C1*C1/GMX
      A=RM/(2.-RM*VM*VM/GMX)
      E=SQRT(1.-P/A)
      CTA=(P-RM)/(E*RM)
      STA=RD*C1/(E*GMX)
      DO 10 I=1,3
      Z(I)= RM/C1*V(I)-RD/C1*R(I)
      PV(I)= CTA*R(I)/RM-STA*Z(I)
   10 Q(I)=STA*R(I)/RM+CTA*Z(I)
C     COMPUTE ROTATION MATRIX
      DO 1 I=1,6
      DO 1 J=1,6
    1 OPEC(I,J)=0.
      DO 11 J=1,3
      OPEC(J,1)=PV(J)
      OPEC(J,2)=Q(J)
   11 OPEC(J,3)=WV(J)
      DO 2 I=1,3
      DO 2 J=1,3
    2 OPEC(I+3,J+3)=OPEC(I,J)
C     COMPUTE TIME FROM  PERIAPSIS
C     COMPUTE TRUE ANOMALY AFTER DELTA TIME
      IF(E-1.)20,20,21
C     ELLIPSE
   20 CSE=(A-RM)/(A*E)
      SNE=RM*STA/(A*SQRT(1.-E*E))
      EA=ATAN2(SNE,CSE)
      ORB=SQRT(A*A*A/GMX)
      TIM1=ORB*(EA-E*SNE)
```

```
      TIM2=TIM1+DELT
      AM2=TIM2/ORB
40    IF(AM2) 41,42,42
41    AM2=AM2+2.*PI
      GO TO 40
42    R3=AM2/6.28318531
      N=R3
      XO=N
      AM2=(R3-XO)*6.28318531
      YO=AM2+E*SIN(AM2)+.5*E*E*SIN(2.*AM2)
      DO 43 I=1,10
      RTHD=SIN(YO)
      DXO=COS(YO)
      R3=YO-E*RTHD
      DYO=(AM2-R3)/(1.-E*DXO)
      IF(ABS(DYO)-.1E-7) 44,43,43
43    YO=YO+DYO
      WRITE(6,4)DYO
    4 FORMAT(13HONO CONV DLE=E15.8)
44    DDYO=RTHD/ABS(RTHD)
      R2=A*(1.-E*DXO)
      CTA2=(P-R2)/(E*R2)
      STA2=SQRT(1.-CTA2*CTA2)*DDYO
      GO TO 22
C     HYPERBOLA
   21 SNF=SQRT(E*E -1.)*STA*RM/P
      F=ALOG(SNF+SQRT(SNF*SNF+1.))
      ORB=SQRT(ABS(A*A*A)/GMX)
      TIM1=ORB*(E*SNF-F)
      TIM2=TIM1+DELT
      AM2=TIM2/ORB
      XO=SIGN(1.,AM2)
      DO 53 I=1,10
      YO=EXP(XO)
      RTHD=(YO-1./YO)*.5
      DXO =(YO+1./YO)*.5
      DYO=-XO+E*RTHD
      R3=(AM2-DYO)/(E*DXO-1.)
      DDXO=ABS(R3)
      IF(DDXO-.1E-7) 54,51,51
51    IF(DDXO-1.) 53,53,52
52    R3=R3/DDXO
53    XO=XO+R3
      WRITE(6,5)R3
5     FORMAT(13HONO CONV DLE=E15.8)
54    R2=A*(1.-E*DXO)
      CTA2=(P-R2)/(E*R2)
      STA2=SQRT(1.-CTA2*CTA2)*SIGN(1.,XO)
C     PSI(T1,0) INVERSE
22    DO 15 I=1,6
      DO 15 J=1,6
15    FMI1(I,J)=0.
      XO=RM*CTA
      YO=RM*STA
      RTHD=C1/RM
      DXO=RD*CTA-RTHD*STA
      DYO=RD*STA+RTHD*CTA
      R3=-GMX/(RM*RM*RM)
      DDXO=XO*R3
      DDYO=YO*R3
```

```
      A1=A/GMX
      A3=1./C1
      A2=A1*A3
      FMI1(1,1) =  A1 *( -DXO - 3.0 * TIM1 * DDXO )
      FMI1(1,2) =  A1 *( -DYO - 3.0 * TIM1 * DDYO )
      FMI1(1,4) = -A1 *( 2.0 * XO - 3.0 * TIM1 * DXO )
      FMI1(1,5) = -A1 *( 2.0 * YO - 3.0 * TIM1 * DYO )
      FMI1(2,1)=A2*((DYO*DYO)+YO*DDYO)
      FMI1(2,2) =  A2 *( -DXO * DYO - YO * DDXO )
      FMI1(2,4) = -A2 * YO * DYO
      FMI1(2,5) = -A2 *( -YO * DXO - 2.0 * C1 )
      FMI1(3,3) = -A3 * DXO
      FMI1(3,6) =  A3 * XO
      FMI1(4,1) = -A1 * DDXO
      FMI1(4,2) = -A1 * DDYO
      FMI1(4,4) =  A1 * DXO
      FMI1(4,5) =  A1 * DYO
      FMI1(5,1) = -A2 *( DYO * DXO + YO * DDXO )
      FMI1(5,2)=-A2*(-(DXO*DXO)-XO*DDXO)
      FMI1(5,4) =  A2 *( YO * DXO - C1 )
      FMI1(5,5) = -A2 * XO * DXO
      FMI1(6,3) = -A3 * DYO
      FMI1(6,6) =  A3 * YO
C     PSI(T2,0)
      DO 16 I=1,6
      DO 16 J=1,6
16    FM1(I,J)=0.
      XO=R2*CTA2
      YO=R2*STA2
      RTHD=C1/R2
      RD=E*STA2*C1/P
      DXO=RD*CTA2-RTHD*STA2
      DYO=RD*STA2+RTHD*CTA2
      R3=-GMX/(R2*R2*R2)
      DDXO=XO*R3
      DDYO=YO*R3
      FM1(1,1) = DXO
      FM1(1,2) = YO * DXO - C1
      FM1(1,4) = 2.0 * XO - 3.0 * TIM2 * DXO
      FM1(1,5) = YO * DYO
      FM1(2,1) = DYO
      FM1(2,2) = -XO * DXO
      FM1(2,4) = 2.0 * YO - 3.0 * TIM2 * DYO
      FM1(2,5) = -YO * DXO - 2.0 * C1
      FM1(3,3) = YO
      FM1(3,6) = -XO
      FM1(4,1) = DDXO
      FM1(4,2) = DYO * DXO + YO * DDXO
      FM1(4,4) = -DXO - 3.0 * TIM2 * DDXO
      FM1(4,5) =(DYO*DYO) + YO * DDYO
      FM1(5,1) = DDYO
      FM1(5,2) =-(DXO*DXO) - XO * DDXO
      FM1(5,4) = -DYO - 3.0 * TIM2 * DDYO
      FM1(5,5) = -DXO * DYO - YO * DDXO
      FM1(6,3) = DYO
      FM1(6,6) = -DXO
C     PSI(T2,T1)=PSI(T2,0)*PSI(T1,0)  (INVERSE)
      DO 25 I=1,6
      DO 25 J=1,6
      PSIOP(I,J)=0.
```

```
      DO 25 K=1,6
25    PSIOP(I,J)=PSIOP(I,J)+FM1(I,K)*FMI1(K,J)
C     PSI(ECLIPTIC)=R*PSI*R(TRANSPOSE)
      DO 30 I=1,6
      DO 30 J=1,6
      PSIEC(I,J)=0.
      DO 30 K=1,6
      DO 30 L=1,6
30    PSIEC(I,J)=PSIEC(I,J)+OPEC(I,K)*PSIOP(K,L)*OPEC(J,L)
      RETURN
      END
```

```
      SUBROUTINE CONIC(R,RM,V,VM,A,E,XI,XL,W,TA,PV,Q,GMX,RP,P,WV)
      DIMENSION  V(3),R(3),WV(3),PV(3),Q(3),Z(3)
      WV(1) = R(2)*V(3) - R(3)*V(2)
      WV(2) = R(3)*V(1) - R(1)*V(3)
      WV(3) = R(1)*V(2) - R(2)*V(1)
      C1 = SQRT(WV(1)**2 + WV(2)**2 + WV(3)**2)
      WV(1) = 1.*WV(1)/C1
      WV(2) = 1.*WV(2)/C1
      WV(3) = 1.*WV(3)/C1
      RRD = R(1) *V(1) + R(2) * V(2) + R(3) * V(3)
      RD=RRD/RM
      P=C1**2/GMX
      A=RM/(2.-RM*VM**2/GMX)
      E=SQRT(1.-P/A)
      RP = P/(1.+E)
      XI=ATAN(SQRT(1.-WV(3)**2)/WV(3))
      XL = ATAN2(WV(1),-WV(2))
      CTA=(P-RM)/(E*RM)
      STA=RD*C1/(E*GMX)
      TA = ATAN2(STA,CTA)
      DO 10 I=1,3
      Z(I)= RM/C1*V(I)-RD/C1*R(I)
      PV(I)= CTA*R(I)/RM-STA*Z(I)
   10 Q(I)=STA*R(I)/RM+CTA*Z(I)
      W = ATAN2(PV(3),Q(3))
      RETURN
      END
```

```
      SUBROUTINE CONST(MODE,NDD,NTT,PI,RAD,AU,AUDAY,AUS,CONV,SSG,RP,HHTA
     1 ,ANG1,ANG2,TIM1,TIM2,DDLAT,DDLON,DDIQ,DDLQ,ROT)
      MODE=3
      PI=3.1415926536
      RAD=57.2957795
      AU = 149.5985
      AUDAY = AU/(24.*3600.)*1000000.
      AUS = 149598500.
      CONV = AUS/86400.
      ROT=15.041
        SSG=2.959122083E-04
      IF(NDD-4)100,101,100
  101 DDIQ= -.4091924432
      DDLQ= 0.0
      GO TO 104
  100 IF(NDD-3)102,103,102
  103 DDIQ=0.0
      DDLQ=0.0
      GO TO 104
  102 IF(NDD-5)104,105,104
  105 DDIQ=0.0
      DDLQ=0.0
  104 CONTINUE
      RP=6560.
      IF(MODE-2)30,30,10
   10 IF(NTT-4)14,12,12
   12 ANG1=23.
      ANG2=25.
      TIM1=700.
      TIM2=300.
      HHTA=12.
      DDLAT=28.28
      DDLON=279.5
      GO TO 20
   14 ANG1=17.
      ANG2=8.
      TIM1=500.
      TIM2=100.
      HHTA=3.7
      DDLAT=28.317
      DDLON=279.457
   20 CONTINUE
   30 CONTINUE
  901 FORMAT(/1X,20HCONVERSION CONSTANTS)
  902 FORMAT(1X,3HPI=,E20.13,3X,7HRADIAN=,E20.13,3X,3HAU=,E20.13,3X,
     1 6HAUDAY=,E20.13)
  903 FORMAT(1X,18HPHYSICAL CONSTANTS)
  904 FORMAT(1X,11HMU OF SUN= ,E17.10,3X,15HROTATION RATE= ,F7.3,3X,
     1 11HOBLIQUITY= ,F13.10,3X,17HLONG OF EQUATOR= ,F13.10)
  905 FORMAT(1X,14HLAUNCH PROFILE)
  906 FORMAT(1X,18HLAUNCH SITE LAT = ,F7.3,3X,19HLAUNCH SITE LONG = ,
     1 F7.3,3X,20HPARKING ORBIT RAD = ,F7.2,3X,19HLAUNCH TRUE ANOM = ,
     2 F6.2)
  907 FORMAT(1X,6HANG1= ,F5.1,3X,6HANG2= ,F5.1,3X,6HTIM1= ,F5.1,3X,
     1 6HTIM2= ,F5.1)
      RETURN
      END
```

```
      SUBROUTINE CONVERT(R,PHI,THETA,VEL,GAMMA,SIGMA,X,Y,Z,VX,VY,VZ)
C          R     = DISTANCE
C          PHI   = DECLINATION
C          THETA = RIGHT ASCENSION
C          VEL   = VELOCITY
C          GAMMA = PATH ANGLE
C          SIGMA = AZIMUTH
      CP=COS(PHI)
      SP=SIN(PHI)
      CT=COS(THETA)
      ST=SIN(THETA)
      CG=COS(GAMMA)
      SG=SIN(GAMMA)
      X=R*CP*CT
      Y=R*CP*ST
      Z=R*SP
      B1=VEL*SG
      B2=VEL*CG*SIN(SIGMA)
      B3=VEL*CG*COS(SIGMA)
      VX=B1*CP*CT-B2*ST-B3*SP*CT
      VY=B1*CP*ST+B2*CT-B3*SP*ST
      VZ=B1*SP+B3*CP
      RETURN
      END
```

```
      SUBROUTINE DATA
C
C
C
C     THIS SUBROUTINE READS THE DATA USED IN EACH OF THE MODES OF
C     OPERATION USED IN THE STEAP PROGRAM
C
C     THE FIRST CARD WHICH SHOULD BE READ CONTAINS THE VARIABLE IRUNX
C     WHICH INDICATES HOW MANY DIFFERENT RUNS ARE TO BE MADE. THIS
C     NUMBER SHOULD BE PLACED ON THE CARD ACCORDING TO AN I10 FIELD.
C     --NOTE--THIS NUMBER IS ONLY READ ONCE EACH TIME THE PROGRAM IS
C     INPUT.
C
C     THE NEXT CARD SHOULD CONTAIN THE VARIABLES IPRO AND ITR WHICH
C     GIVE THE PROBLEM IDENTIFICATION AND A CODE NUMBER INDICATING
C     WHICH MODE TO BE USED FOR THIS PROBLEM. THE FORMAT IS 2I10.
C     --NOTE--THIS CARD SHOULD PRECEDE THE INPUT FOR EACH RUN.
C
C        ITR      --  =1  --  TRAJECTORY MODE
C                     =2  --  TARGETING MODE
C                     =3  --  ERROR ANALYSIS MODE
C                     =4  --  SIMULATION MODE
C
C
C     IF THE TRAJECTORY MODE IS TO BE RUN, THE DATA IS INPUT THROUGH
C     THE USE OF A NAMELIST ENTITLED TRAJ WHICH INCLUDES THE FOLLOWING
C     VARIABLES.
C
C        XI       --  A VECTOR CONTAINING THE INITIAL POSITION AND
C                     VELOCITY OF THE VEHICLE
C                     --NOTE--THIS VECTOR IS READ ONLY IF ICOOR = 0,1,2
C        ICOOR    --  A CODE TO DETERMINE WHAT COORDINATE SYSTEM THE
C                     INITIAL STATE VECTOR IS IN.  IF THIS CODE IS NOT
C                     INCLUDED IN THE NAMELIST, IT IS ASSUMED TO BE 2.
C                     =0 -- HELIOCENTRIC ECLIPTIC
C                     =1 -- GEOCENTRIC EQUATORIAL
C                     =2 -- GEOCENTRIC ECLIPTIC
C                     =3 -- JPL CONDITIONS (RDS,PHI,THETA,VEL,GAMMA,
C                           SIGMA)
C        RDS      --  EARTH-CENTERED INJECTION RADIUS
C        PHI      --  DECLINATION
C        THETA    --  INJECTION RIGHT ASCENSION
C        VEL      --  INERTIAL INJECTION SPEED
C        GAMMA    --  INJECTION PATH ANGLE
C        SIGMA    --  INJECTION AZIMUTH
C                     --NOTE--THESE CONDITIONS ARE INPUT ONLY IF
C                     ICOOR = 3.
C        LMO      --  LAUNCH MONTH (INTEGER)
C        LDAY     --  LAUNCH DAYS (INTEGER)
C        LHR      --  LAUNCH HOURS (INTEGER)
C        LMIN     --  LAUNCH MINUTES (INTEGER)
C        SECL     --  LAUNCH SECONDS (FLOATING)
C        LYR      --  LAUNCH YEAR (INTEGER)
C        IMO      --  MONTH OF FINAL COMPUTATION EINTEGER)
C        IDAY     --  DAY OF FINAL COMPUTATION (INTEGER)
C        IHR      --  HOUR OF FINAL COMPUTATION (INTEGER)
C        IMIN     --  MINUTE OF FINAL COMPUTATION (INTEGER)
C        SECI     --  SECOND OF FINAL COMPUTATION (FLOATING)
C        IYR      --  YEAR OF FINAL COMPUTATION (INTEGER)
C        ALNGTH   --  LENGTH UNITS PER A.U.
```

```
C                     --NOTE--IF ALNGTH IS NOT READ IN, THE LENGTH UNITS
C                     ARE ASSUMED TO BE KILOMETERS (ALNGTH=149598500.)
C        TM      --   TIME UNITS PER DAY
C                     --NOTE--IF TM IS NOT READ IN, THE TIME UNITS ARE
C                     ASSUMED TO BE SECONDS. (TM=86400.)
C        NTMC    --   NOMINAL TRAJECTORY MODULE CODE
C                     =1 -- PATCHED CONIC
C                     =2 -- VIRTUAL MASS
C                     --NOTE--IF NOT INPUT, CODE 2 IS ASSUMED
C        TRTM1   --   INITIAL TRAJECTORY TIME (ASSUMED ZERO IF NOT
C                     READ IN)
C        NBOD    --   NUMBER OF BODIES TO BE CONSIDERED IN ANALYSIS
C                     --NOTE--IF NBOD IS NOT INPUT, IT IS ASSUMED TO BE 3
C                     (SUM, LAUNCH PLANET, TARGET PLANET)
C        NB      --   ARRAY OF CODES OF BODIES
C                     =1 -- SUN
C                     =2 -- MERCURY
C                     =3 -- VENUS
C                     =4 -- EARTH
C                     =5 -- MARS
C                     =6 -- JUPITER
C                     =7 -- SATURN
C                     =8 -- URANUS
C                     =9 -- NEPTUNE
C                     =10-- PLUTO
C                     =11-- EARTHS MOON
C                     --NOTE--NB(2)=LAUNCH PLANET
C                             NB(3)=TARGET PLANET
C      THE FOLLOWING INFORMATION IS NECESSARY ONLY IF NTMC=2.
C        IEPHEM  --   EPHEMERIS CODE
C                     =0  --  PLACE EACH PLANET IN ELLIPSE
C                             THE DATE AT WHICH THIS ELLIPSE IS
C                             CALCULATED IS DETERMINED BY READING IN A
C                             VARIABLE WHICH IS ENTITLED AS THE NAME OF
C                             THE PLANET CONSIDERED.  THIS VARIABLE
C                             MONTH, DAY, HOUR, MINUTE,SECOND, AND
C                             SHOULD CONTAIN SIX NUMBERS SPECIFYING THE
C                             YEAR.  (EXAMPLE..EARTH=7,24,6,15,38,1973)
C                             --NOTE--IF THESE VARIABLES ARE OMITTED FROM
C                             THE NAMELIST THE FOLLOWING RULES WILL BE
C                             APPLIED.
C                             NB(2) AT LAUNCH DATE
C                             NB(3) AT TARGET DATE
C                             MOON AT SAME DATE AS EARTH
C                             ALL OTHERS AT TARGET DATE
C                     =1  --  CALCULATE ORBITAL ELEMNETS FOR EACH PLANET
C                             EACH TIME INTERVAL
C                     --NOTE--ONE  IS ASSUMED IF THIS VARIABLE IS NOT
C                     INPUT
C        IPRINT  --   PRINT CODE FOR VIRTUAL MASS
C                     =0  ALL OUTPUT WILL BE PRINTED BOTH INITIALLY AND
C                         FINALLY AS USUAL
C                         (ASSUMED IF NOT INPUT IN TRAJECTORY MODE)
C                     =1  OUTPUT WILL BE SUPPRESSED AT THE BEGINNING AND
C                         FINAL STEPS
C                         (ASSUMED IF NOT INPUT IN ERROR ANALYSIS MODE
C                         AND SIMULATION MODE)
C        ISP2    --   CODE FOR VIRTUAL MASS TRAJECTORY
C                     =0      CONTINUE INTEGRATING TO FINAL TIME
C                             --NOTE--ISP2 IS ASSUMED ZERO IF NOT INPUT
```

```
C                    .GT.0  STOP INTEGRATING WHEN SPHERE OF INFLUENCE
C                           OF STATED PLANET IS ENCOUNTERED.
C        ACC     --  ACCURACY FIGURE  (ASSUMED 1.E-6 IF NOT INPUT
C        DELTP   --  PRINT INTERVAL (IN DAYS)--ASSUMED 3. IF NOT INPUT
C                    IN TRAJECTORY MODE.  IF NOT INPUT IN ERROR ANALYSIS
C                    MODE OR SIMULATION MODE DELTP IS ASSUMED 500.
C        INPR    --  PRINT INTERVAL (INCREMENTS) (ASSUMED 100 IF NOT
C                    INPUT IN TRAJECTORY MODE.  IN ERROR ANALYSIS OR
C                    SIMULATION MODE ASSUMED 10000 IF NOT INPUT)
C
C     IF THE TARGETING MODE IS TO BE RUN, A NAMELIST -TARG- IS TO BE
C     READ WITH THE FOLLOWING VARIABLES.
C
C     IF THE ERROR ANALYSIS MODE IS TO BE RUN, A NAMELIST ENTITLED
C     ERRAN IS READ WHICH INCLUDES ALL OF THOSE VARIABLES USED IN THE
C     TRAJECTORY MODE PLUS THE FOLLOWING.
C
C        IAUG    --  AUGMENTATION FLAG
C                    = 1   STATE VECTOR CONSISTS OF POSITION AND
C                          VELOCITY OF VEHICLE (NDIM = 6)
C
C                --  ALL REMAINING CODES INDICATE STATE VECTORS WITH
C                    AUGMENTED INFORMATION AS NOTED  --
C
C                    = 2   STATION LOCATION PARAMETERS (NDIM = 9)
C                          (GEOCENTRIC RADIUS, LATITUDE, LONGITUDE)
C                    = 3   MU OF SUN AND MU OF TARGET PLANET (NDIM = 8)
C                    = 4   SIX MEASUREMENT BIASES (RANGE BIAS, RANGE-
C                          RATE BIAS, THREE STAR ANGLE BIASES, APPARENT
C                          PLANET DIAMETER BIAS)  (NDIM=12)
C                    = 5   THREE EPHEMERIS BIASES OF TARGET PLANET
C                          (SEMI-MAJOR AXIS BIAS, ECCENTRICITY BIAS,
C                    = 6   NINE STATION LOCATION PARAMETERS (NDIM=15)
C                          (THREE FROM EACH OF THREE STATIONS)
C                    = 7   THREE STATION LOCATION PARAMETERS PLUS MU OF
C                          SUN AND MU OF TARGET PLANET (NDIM = 11)
C                    = 8   THREE STATION LOCATION PARAMETERS AND SIX
C                          MEASUREMENT BIASES  (NDIM=15)
C                    = 9   MU OF SUN, MU OF TARGET PLANET, THREE
C                          EPHEMERIS BIASES  (NDIM=11)
C                    = 10  SIX MEASUREMENT BIASES AND THREE EPHEMERIS
C                          BIASES  (NDIM=15)
C                    = 11  THREE STATION LOCATION PARAMETERS PLUS MU
C                          OF SUN, MU OF PLANET, SIX MEASUREMENT
C                          BIASES (NDIM=17)
C
C        NENT    --  NUMBER OF ENTRIES IN THE MEASUREMENT SCHEDULE
C                    THIS IS ASSUMED ZERO IF IT IS NOT INPUT
C                    --NOTE--THE MEASUREMENT SCHEDULE ITSELF IS NOT
C                    READ IN THE NAMELIST.  IT WILL BE READ
C                    IMMEDIATLEY FOLLOWING THE NAMELIST
C                    SCHED(1)   SCHED(2)   SCHED(3)   MEAS
C                      F10.0      F10.0      F10.0    I10
C                    DAY1    TO  DAY2  EVERY X DAYS W. CODE
C                    THE CODE IS DETERMINED BY THE FOLLOWING LIST
C                    =1  RANGE-RATE --  IDEALIZED STATION
C                    =2  RANGE, RANGE-RATE -- IDEALIZED STATION
C                    =3  RANGE-RATE -- STATION 1
C                    =4  RANGE, RANGE-RATE --  STATION 1
C                    =5  RANGE-RATE -- STATION 2
```

```
C                    =6  RANGE, RANGE-RATE --  STATION 2
C                    =7  RANGE-RATE -- STATION 3
C                    =8  RANGE, RANGE-RATE --  STATION 3
C                    =9  THREE STAR PLANET ANGLES
C                    =10 APPARENT PLANET DIAMETER
C       NEV1    --   NUMBER OF EIGENVECTOR EVENTS (ASSUMED ZERO, IF
C                    NOT READ)
C       T1      --   ARRAY OF TIMES AT WHICH EIGENVECTOR EVENTS OCCUR
C                    --NOTE--THIS IS TO BE INPUT ONLY IF NEV1.NE.0
C       IEIG    --   EIGENVECTOR CODE
C                    =0 -- ONLY POSITION EIGENVECTORS WILL BE INPUT
C                          (ASSUMED IF NOT INPUT)
C                    =1 -- BOTH POSITION AND VELOCITY EIGENVECTORS
C                          WILL BE CALCULATED
C       IHYP1   --   HYPERELLPSOID SIGMA LEVEL CODE
C                    =1 -- SIGMA LEVEL EQUALS ONE
C                    =2  - SIGMA LEVEL EQUALS THREE (ASSUMED IF NOT
C                          INPUT)
C                    =3 -- SIGMA LEVEL OF BOTH ONE AND THREE
C       NEV2    --   NUMBER OF PREDICTION EVENTS (ASSUMED ZERO, IF
C                    NOT READ)
C       T2      --   ARRAY OF TIMES AT WHICH PREDICTION EVENTS OCCUR
C                    --NOTE--THIS IS INPUT ONLY IF NEV2.NE.0
C       TPT2    --   ARRAY OF TIMES TO WHICH ONE WISHES TO PREDICT.
C                    --NOTE--THESE MUST CORRESPOND TO THOSE TIMES
C                    LISTED IN T2 AND SHOULD BE INPUT ONLY IF T2 IS
C                    INPUT
C       NEV3    --   NUMBER OF GUIDANCE EVENTS (ASSUMED TO BE ZERO IF
C                    NOT INPUT)
C       T3      --   ARRAY OF TIMES AT WHICH GUIDANCE EVENTS OCCUR
C                    --NOTE--THESE MUST BE INPUT ONLY IF NEV3.NE.0
C       ICDT3   --   ARRAY OF CODES WHICH DETERMINE WHAT GUIDANCE
C                    POLICY IS TO BE USED AT EACH GUIDANCE EVENT
C                    =1 -- FIXED TIME OF ARRIVAL
C                    =2 -- TWO-VARIABLE B-PLANE
C                    =3 -- THREE-VARIABLE B-PLANE
C                    --NOTE--THESE CODES MUST CORRESPOND TO THE TIMES
C                    AS STATED IN T3 AND NEED BE INPUT ONLY IF T3 IS
C                    INPUT--IF THESE ARE NOT INPUT WHEN T3 IS INPUT
C                    THREE VARIABLE B-PLANE IS ASSUMED
C       ICDQ3   --   ARRAY OF CODES FOR GUIDANCE EVENTS TO DETERMINE
C                    HOW THE EXECUTION ERROR IS TO BE CALCULATED.
C                    = 0  CALCULATED DIRECTLY FROM S MATRIX
C                    = 1  CALCULATED FROM EIGENVECTOR CORRESPONDING TO
C                         MAXIMUM EIGENVALUE OF S MATRIX.
C                    --NOTE--THESE CODES MUST CORRESPOND TO THE TIMES AS
C                    STATED IN T3 AND NEED BE INPUT ONLY IF T3 IS INPUT.
C                    IF THESE ARE NOT INPUT WHEN T3 IS INPUT, OPTION 1
C                    IS ASSUMED
C       SIGRES  --   VARIANCE OF RESOLUTION ERROR
C                    ASSUMED 4.E-8 KM**2/SEC**2 IF NOT INPUT
C       SIGPRO  --   VARIANCE OF PROPORTIONALITY ERROR
C                    ASSUMED .0001 IF NOT INPUT
C       SIGALP  --   VARIANCE OF POINTING ANGLE ALPHA
C                    ASSUMED .0043625 RADIANS IF NOT INPUT
C       SIGBET  --   VARIANCE OF POINTING ANGLE BETA
C                    ASSUMED .0043625 RADIANS IF NOT INPUT
C                    --NOTE--THE ABOVE SIGMA VALUES MUST BE INPUT ONLY
C                    IF NEV3.NE.0
C       P       --   INITIAL COVARIANCE MATRIX.
```

```
C                      --NOTE--IF THIS MATRIX IS NOT INPUT, A DIAGONAL
C                      MATRIX IS ASSUMED WITH THE LISTED VALUES FOR THE
C                      FIRST SIX ELEMENTS ON THE DIAGONAL.  ALL OTHERS
C                      WILL BE ZERO
C                      1.    1.    1.    1.E-4    1.E-4    1.E-4
C          ISTMC   --  STATE TRANSITION MATRIX CODE
C                      =1 -- PATCHED-CONIC (ANALYTICAL)
C                      =2 -- VIRTUAL MASS (ANALYTICAL)
C                      =3 -- NUMERICAL DIFFERENCING USING VIRTUAL-MASS.
C                      --NOTE--IF THIS CODE IS NOT INPUT,ISTMC = 1 IS
C                      ASSUMED.
C          IDNF    --  DYNAMIC NOISE FLAG
C                      =0 -- DYNAMIC NOISE IS ZERO--ASSUMED IF NOT INPUT
C                      =1 -- DYNAMIC NOISE IS NOT ZERO
C          DNCN    --  CONSTANTS USED TO CALCULATE DYNAMIC NOISE (NNED BE
C                      INPUT ONLY IF IDNF=1)
C          IMNF    --  MEASUREMENT NOISE FLAG
C                      =0 -- MEASUREMENT NOISE IS CONSTANT (ASSUMED IF
C                            NOT READ IN)
C                      =1 -- MEASUREMENT NOISE IS NOT CONSTANT
C          MNCN    --  ARRAY OF VARIANCES FOR EACH TYPE OF MEASUREMENT
C                      --NOTE--IF THIS ARRAY IS OMITTED FROM THE NAMELIST,
C                      THE FOLLOWING ARRAY WILL BE ASSUMED.
C                      RANGE (IDEALIZED STATION)       1.E-6
C                      RANGE-RATE (IDEALIZED STATION)  1.E-12
C                      RANGE (STATION 1)               1.E-6
C                      RANGE-RATE (STATION 1)          1.E-12
C                      RANGE (STATION 2)               1.E-6
C                      RANGE-RATE (STATION 2)          1.E-12
C                      RANGE (STATION 3)               1.E-6
C                      RANGE-RATE (STATION 3)          1.E-12
C                      STAR ANGLE 1                    2.5E-9
C                      STAR ANGLE 2                    2.5E-9
C                      STAR ANGLE 3                    2.5E-9
C                      APPARENT PLANET DIAMETER        2.5E-9
C          NST     --  NUMBER OF TRACKING STATIONS ON THE ROTATING EARTH
C                      --NOTE--THIS INFORMATION NEED BE INPUT ONLY IF
C                      INCLUDED IN THE MEASUREMENTS IS TYPE 3,4,5,6,7,8
C                      IF NO INFORMATION ON THE TRACKING STATIONS IS INPUT
C                      THREE STATIONS WILL BE ASSUMED AS THE FOLLOWING
C                                           ALT         LAT        LONG
C                      1.  GOLDSTONE   --  1.031 KM   35.384N   116.833W
C                      2.  MADRID      --   .050 KM   40.417N     3.667W
C                      3.  CANBERRA    --   .050 KM   35.311S   149.136E
C          SAL     --  ARRAY OF ALTITUDES OF EACH TRACKING STATION
C          SLAT    --  ARRAY OF LATITUDES OF EACH TRACKING STATION
C          SLON    --  ARRAY OF LONGITUDES OF EACH TRACKING STATION
C                      --NOTE--THE ABOVE THREE ARRAYS MUST BE INPUT ONLY
C                      IF NST.NE.0
C          U1,V1,W1--  DIRECTION COSINES OF STAR PLANET ANGLE 1
C                      (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
C                      IF THESE ARE NOT INPUT STAR NUMBER 1 IS ASSUMED
C                      TO BE CANOPUS WITH
C                      U1=-.061351,  V1=.237886,  W1=-.969355
C          U2,V2,W2--  DIRECTION CONSINES OF STAR PLANE ANGLE 2
C                      (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
C                      IF THESE ARE NOT INPUT STAR NUMBER 2 IS ASSUMED
C                      TO BE BETELGEUSE WITH
C                      U2=.028986,  V2=.960388,  W2=-.277141
C          U3,V3,W3--  DIRECTION COSINES OF STAR PLANET ANGLE 3
```

```
C                    (NECESSARY ONLY IF THIS ANGLE IS BEING MEASURED)
C                    IF THESE ARE NOT INPUT STAR NUMBER 3 IS ASSUMED
C                    TO BE RIGEL WITH
C                    W3=.201963, V3=.831343, W3=-.517784
C        FACP    --  POSITION FACTOR FOR NUMERICAL DIFFERENCING
C                    (NEED BE INPUT ONLY IF ISTMC=3)
C                    ASSUMED TO BE 1 KM IF NOT INPUT
C        FACV    --  VELOCITY FACTOR FOR NUMERICAL DIFFERENCING
C                    (NEED BE INPUT ONLY IF ISTMC=3)
C                    ASSUMED TO BE 1.E-4 KM/SEC IF NOT INPUT.
C        FOP     --  A VALUE TO BE USED AS AN OFF-DIAGONAL ANIHILATION
C                    ELEMENT IN THE EIGENVECTOR ROUTINE FOR POSITION
C                    EIGENVALUES AND EIGENVECTORS (ASSUMED TO BE 1.E-15
C                    IF NOT READ IN)
C        FOV     --  A VALUE TO BE USED AS AN OFF-DIAGONAL ANIHILATION
C                    ELEMENT IN THE EIGENVECTOR ROUTINE FOR VELOCITY
C                    EIGENVALUES AND EIGENVECTORS (ASSUMED TO BE 1.E-25
C                    IF NOT READ IN)
C        ISTM1   --  AN ALTERNATE STATE TRANSITION MATRIX CODE
C                    =0  IF DELTM IS GREATER THAN DTMAX (DESCRIBED
C                        BELOW) CALCULATE PSI BY USING THE SUN AS THE
C                        GOVERNING BODY.  (ASSUMED 3 IF NOT INPUT)
C                    =1  IF DELTM IS GREATER THAN DTMAX CALCULATE PSI BY
C                        NUMERICAL DIFFERENCING.
C        DTMAX   --  THE MAXIMUM DELTM (IN DAYS) SO THAT THE STATE
C                    TRANSITION MATRIX COMPUTATION IS CONSIDERED VALID
C                    WHEN USING EITHER THE PATCHED CONIC TECHNIQUE OR
C                    THE VIRTUAL MASS TECHNIQUE
C                    (ASSUMED TO BE 8 DAYS IF NOT READ IN)
C        NDACC   --  ACCURACY CODE FOR NUMERICAL DIFFERENCING
C                    =0  USE THE SAME ACCURACY FIGURE IN THE CALCULATION
C                        OF THE STATE TRANSITION MATRIX BY THE METHOD OF
C                        NUMERICAL DIFFERENCING AS IS USED IN THE
C                        NOMINAL TRAJECTORY (ASSUMED IF NOT INPUT)
C                    =1  CHANGE THE ACCURACY IN USING THE NUMERICAL
C                        DIFFERENCING METHOD TO ACCND (DESCRIBED BELOW)
C        ACCND   --  ACCURACY TO BE USED IN THE CALCULATION OF THE STATE
C                    TRANSITION MATRIX BY THE METHOD OF NUMERICAL
C                    DIFFERENCING.  (USED ONLY IF NDACC=1) ASSUMED TO BE
C                    2.5E-5 IF NOT INPUT.
C        DELAXS  --  SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL
C                    DIFFERENCING TO COMPUTE PSI AND H IF IAUG = 5, 9,
C                    OR 10.  (ASSUMED 100 KM IF NOT INPUT)
C        DELECC  --  ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING
C                    TO COMPUTE PSI AND H IF IAUG = 5, 9, OR 10.
C                    (ASSUMED 1.E-5 IF NOT INPUT)
C        DELICL  --  INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING
C                    TO COMPUTE PSI AND H IF IAUG = 5, 9, OR 10.
C                    (ASSUMED 10 ARCSECONDS IF NOT INPUT)
C        DELMUS  --  FACTOR USED IN NUMERICAL DIFFERENCING FOR THE MU OF
C                    THE SUN TO GENERATE THE AUGMENTED STATE TRANSITION
C                    MATRIX WHEN IAUG = 3, 7, 9, OR 11.  (ASSUMED 1.E7
C                    WHEN NOT INPUT)
C        DELMUP  --  FACTOR USED IN NUMERICAL DIFFERENCING FOR THE MU OF
C                    THE TARGET PLANET TO GENERATE THE AUGMENTED STATE
C                    TRANSITION MATRIX WHEN IAUG = 3, 7, 9, OR 11.
C                    (ASSUMED .1 WHEN NOT INPUT)
C
C     IN ORDER TO EXERCISE THE SIMULATION OPTION, A NAMELIST ENTITLED
C     SMLTN IS READ WHICH CONTAINS ALL THE VARIABLES MENTIONED ABOVE
```

```
C      FOR THE TRAJECTORY AND ERROR ANALYSIS MODES PLUS THE  FOLLOWING.
C
C         NEV4      --   NUMBER OF QUASI-LINEAR FILTERING EVENTS TO BE RUN
C         T4        --   AN ARRAY OF TIMES AT WHICH QUASI-LINEAR FILTERING
C                        EVENTS ARE TO TAKE PLACE.
C                        --NOTE--THIS ARRAY IS NECESSARY ONLY IF NEV4 IS NOT
C                        ZERO
C         ADEVX     --   THE VECTOR DESCRIBING THE ACTUAL DEVIATION OF THE
C                        ACTUAL TRAJECTORY FROM THE MOST RECENT NOMINAL
C                        TRAJECTORY
C         BIA       --   AN ARRAY OF MEASUREMENT BIASES WHICH DETERMINE THE
C                        ACTUAL VALUE TO BE USED FOR EACH OF THE TYPES OF
C                        MEASUREMENTS
C         DMUSB     --   ACTUAL BIAS OF THE MU OF THE SUN TO BE USED IN THE
C                        DETERMINATION OF THE ACTUAL TRAJECTORY
C                        (ASSUMED TO BE ZERO IF NOT INPUT)
C         DMUPB     --   ACTUAL BIAS OF THE MU OF THE TARGET PLANET TO BE
C                        USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
C                        (ASSUMED TO BE ZERO IF NOT INPUT)
C         DAB       --   ACTUAL BIAS IN THE SEMI-MAJOR AXIS OF THE TARGET
C                        PLANET TO BE USED IN THE DETERMINATION OF THE
C                        ACTUAL TRAJECTORY
C                        (ASSUMED TO BE ZERO IF NOT INPUT)
C         DEB       --   ACTUAL BIAS IN THE ECCENTRICITY OF THE TARGET
C                        PLANET TO BE USED IN THE DETERMINATION OF THE
C                        ACTUAL TRAJECTORY
C                        (ASSUMED TO BE ZERO IF NOT INPUT)
C         DIB       --   ACTUAL BIAS IN THE INCLINATION OF THE TARGET PLANET
C                        TO BE USED IN THE DETERMINATION OF THE ACTUAL
C                        TRAJECTORY
C                        (ASSUMED TO BE ZERO IF NOT INPUT)
C         TTIM1     --   THE FIRST TIME AT WHICH THE VALUES USED FOR THE
C                        ACTUAL UNMODELLED ACCELERATION WILL BE ALTERED
C         TTIM2     --   THE SECOND TIME AT WHICH THE VALUES USED FOR THE
C                        ACTUAL UNMODELLED ACCELERATION WILL BE ALTERED
C         UNMAC     --   AN ARRAY OF VALUES WHICH DETERMINE THE ACTUAL
C                        UNMODELLED ACCELERATION TO BE USED AT A GIVEN TIME
C                        NOTE--THESE VALUES ARE ASSUMED ZERO IF NOT INPUT
C                        T0 - T1      T1 - T2      T2 - TF
C                          X1           X2           X3     ACCELERATION
C                          Y1           Y2           Y3     ACCELERATION
C                          Z1           Z2           Z3     ACCELERATION
C         SLB       --   AN ARRAY OF ACTUAL BIASES IN THE LOCATIONS OF THE
C                        THREE ROTATING STATIONS ON THE EARTH
C                        (AL1,LAT1,LONG1,AL2,LAT2,LONG2,AL3,LAT3,LONG3)
C                        NOTE--THESE VALUES ARE ASSUMED TO BE ZERO IF NOT
C                        INPUT
C         IAMNF     --   ACTUAL MEASUREMENT NOISE CODE
C                        =0  --  ASSUME THE ACTUAL UNCERTAINTIES IN THE
C                                MEASUREMENT NOISE ARE THE SAME AS THE
C                                UNCERTAINTIES ASSUMED IN THE MOST RECENT
C                                NOMINAL TRAJECTORY
C                        =1  --  CALCULATE THE ACTUAL UNCERTAINITES IN THE
C                                MEASUREMENT NOISE USING THE FOLLOWING
C                                CONSTANTS
C                        NOTE  --  IF NOT INPUT IAMNF IS ASSUMED ZERO
C         AVARM     --   ACTUAL VARIANCES TO BE USED IN COMPUTING THE ACTUAL
C                        UNCERTAINTIES IN THE MEASUREMENT FROM WHICH THE
C                        ACTUAL MEASUREMENT NOISE IS CALCULATED
C         NBOD1     --   NUMBER OF BODIES TO BE CONSIDERED IN THE ACTUAL
```

```
C                          TRAJECTORY (ASSUMED TO BE 11 IF NOT INPUT)
C       NB1        --      AN ARRAY OF CODES OF PLANETS TO BE USED IN THE
C                          ACTUAL TRAJECTORY (IF NOT INPUT ALL MAJOR PLANETS
C                          IN THE SOLAR SYSTEM ARE CONSIDERED PLUS THE SUN AND
C                          THE EARTHS MOON)
C       ACC1       --      AN ACCURACY FIGURE TO BE USED IN THE VIRTUAL MASS
C                          PROGRAM WHEN GENERATING THE ACTUAL TRAJECTORY
C                          (IF NOT INPUT ACC1 IS ASSUMED TO BE 1.E-6)
C       ARES       --      AN ARRAY OF ACTUAL RESOLUTION ERRORS CORRESPONDING
C                          TO THE GUIDANCE EVENTS.  (ASSUMED 3 IF NOT INPUT)
C                          --NOTE--NEED BE INPUT ONLY IF GUIDANCE EVENTS OCCUR
C       APRO       --      AN ARRAY OF ACTUAL PROPORTIONALITY ERRORS FOR EACH
C                          GUIDANCE EVENT.  (ASSUMED ZERO IF NOT INPUT)
C                          --NOTE--NEED BE INPUT ONLY IF GUIDANCE EVENTS OCCUR
C       AALP       --      AN ARRAY OF ACTUAL ERRORS FOR POINTING ANGLE ONE
C                          FOR THE GUIDANCE EVENTS.  (ASSUMED ZERO IF NOT
C                          INPUT)  --NOTE--NEED BE INPUT ONLY IF GUIDANCE
C                          EVENTS OCCUR
C       ABET       --      AN ARRAY OF ACTUAL ERRORS FOR POINTING ANGLE TWO
C                          FOR THE GUIDANCE EVENTS.  (ASSUMED ZERO IF NOT
C                          INPUT)  --NOTE--NEED BE INPUT ONLY IF GUIDANCE
C                          EVENTS OCCUR
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     *,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON /MEAS/ TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DIMENSION SCHED(50,3),MEAS(50),AP(50),DATE(11)
      DIMENSION T1(20),T2(20),T3(20),T4(20)
      DIMENSION MERCURY(6),VENUS(6),EARTH(6),MOON(6),MARS(6),SATURN(6),
```

```
$URANUS(6),NEPTUNE(6),PLUTO(6),JUPITER(6)
 INTEGER VENUS,EARTH,SATURN,URANUS,PLUTO
 REAL MNCN
 DATA MDNM/10HTRAJECTORY,10HTARGETING ,10HERROR ANAL,
$10HSIMULATION,10H MODE     ,10HMODE      ,10HYSIS MODE ,
$10H MODE     /
 DATA EVNM/10HEIGENVALUE,10HPREDICTION,10HGUIDANCE  ,10HQ-LINEAR  /
 DATA  MNNAME/10HRANGE (EAR,10HRANGE-RATE,10HRANGE (STA,
$10HRANGE-RATE,10HRANGE (STA,10HRANGE-RATE,10HRANGE (STA,
$10HRANGE-RATE,10HSTAR PLANE,10HSTAR PLANE,10HSTAR PLANE,
$10HAPPARENT P,10HTH-CENTERE,10H (EARTH-CE,10HTION NUMBE,
$10H (STATION ,10HTION NUMBE,10H (STATION ,10HTION NUMBE,
$10H (STATION ,10HT ANGLE NU,10HT ANGLE NU,10HT ANGLE NU,
$10HLANET DIAM,10HD. . . . .,10HNTERED . .,10HR 1) . . .,
$10HNUMBER 1). ,10HR 2) . . .,10HNUMBER 2).,10HR 3) . . .,
$10HNUMBER 3).,10HMBER 1 . .,10HMBER 2 . .,10HMBER 3 . .,
$10HETER . . ./
 DATA CMPNM/11*10HRX        ,11*10HRY        ,11*10HRZ        ,11*
$10HVX        ,11*10HVY        ,11*10HVZ        ,10H        ,10HR
$ADIUS 1  ,10HMU OF SUN ,10HRANGE BIAS,10HA BIAS    ,3*10HRADIUS 1
$ ,10HMU OF SUN ,10HRANGE BIAS,10HRADIUS 1  ,10H        ,10HLATIT
$UDE 1,10HMU-TARG PL,10HR-RAT BIAS,10HECC BIAS  ,3*10HLATITUDE 1,10
$HMU-TARG PL,10HR-RAT BIAS,10HLATITUDE 1,10H        ,10HLONG 1
$ ,10H        ,10HSTAR ANG 1,10HINC BIAS  ,3*10HLONG 1    ,10HA B
$IAS    ,10HSTAR ANG 1,10HLONG 1    ,3*10H        ,10HSTAR ANG 2,
$10H        ,10HRADIUS 2  ,10HMU OF SUN ,10HRANGE BIAS,10HECC BIA
$S  ,10HSTAR ANG 2,10HMU OF SUN ,3*10H        ,10HSTAR ANG 3,10H
$        ,10HLATITUDE 2,10HMU-TARG PL,10HR-RAT BIAS,10HINC BIAS  ,
$10HSTAR ANG 3,10HMU-TARG PL,3*10H        ,10HAPP DIAM  ,10H
$     ,10HLONG 2    ,10H        ,10HSTAR ANG 1,10H        ,10HA
$PP DIAM  ,10HRANGE BIAS,5*10H        ,10HRADIUS 3  ,10H
$ ,10HSTAR ANG 2,10H        ,10HA BIAS    ,10HR-RAT BIAS,5*10H
$        ,10HLATITUDE 3,10H        ,10HSTAR ANG 3,10H        ,10
$HECC BIAS  ,10HSTAR ANG 1,5*10H        ,10HLONG 3    ,10H
$    ,10HAPP DIAM  ,10H        ,10HINC BIAS  ,10HSTAR ANG 2,10*10H
$          ,10HSTAR ANG 3,10*10H        ,10HAPP DIAM  /
 NAMELIST /TRAJ/ XI,ICOOR,LMO,LDAY,LHR,LMIN,SECL,LYR,IMO,IDAY,IHR,
$IMIN,SECI,IYR,ALNGTH,TM,NTMC,NBOD,NB,ACC,DELTP,INPR,TRTM1,RDS,PHI,
$THETA,VEL,GAMMA,SIGMA,ISP2,IEPHEM,MERCURY,VENUS,EARTH,MARS,SATURN,
$URANUS,NEPTUNE,PLUTO,MOON,JUPITER,IPRINT
 NAMELIST /ERRAN/ XI,ICOOR,IAUG,LMO,LDAY,LHR,LMIN,SECL,LYR,IMO,
$IDAY,IHR,IMIN,SECI,IYR,ALNGTH,TM,NTMC,NBOD,NB,ACC,DELTP,INPR,NENT,
$NEV1,NEV2,NEV3,T1,T2,TPT2,T3,ICDT3,P,ISTMC,IDNF, DNCN,IMNF,MNCN,
$NST,SAL,SLAT,SLON,IEIG,IHYP1,TRTM1,RDS,PHI,THETA,VEL,GAMMA,
$SIGMA,U1,U2,U3,V1,V2,V3,W1,W2,W3,FACP,FACV,ISP2,ICDQ3,SIGRES,SIGPR
$O,SIGALP,SIGBET,FOP,IEPHEM,MERCURY,VENUS,EARTH,MARS,JUPITER,SATURN
$,URANUS,NEPTUNE,PLUTO,MOON,IPRINT,FOV,ISTM1,DTMAX,NDACC,ACCND
$,DELAXS,DELECC,DELICL,DELMUP,DELMUS
 NAMELIST /SMLTN/ XI,ICOOR,IAUG,LMO,LDAY,LHR,LMIN,SECL,LYR,IMO,
$IDAY,IHR,IMIN,SECI,IYR,ALNGTH,TM,NTMC,NBOD,NB,ACC,DELTP,INPR,NENT,
$NEV1,NEV2,NEV3,T1,T2,TPT2,T3,ICDT3,P,ISTMC,IDNF, DNCN,IMNF,MNCN,
$NST,SAL,SLAT,SLON,IEIG,IHYP1,TRTM1,RDS,PHI,THETA,VEL,GAMMA,
$SIGMA,U1,U2,U3,V1,V2,V3,W1,W2,W3,FACP,FACV,ISP2,ICDQ3,SIGRES,SIGPR
$O,SIGALP,SIGBET,FOP,IEPHEM,MERCURY,VENUS,EARTH,MARS,JUPITER,SATURN
$,URANUS,NEPTUNE,PLUTO,MOON,IPRINT,FOV,ISTM1,DTMAX,NDACC,ACCND
$,DELAXS,DELECC,DELICL,DELMUP,DELMUS,BIA,NEV4,T4,ADEVX,DMUSB,DMUPB,
$TTIM1,TTIM2,UNMAC,SLB,DIB,IAMNF,AVARM,NBOD1,NB1,ACC1,DAB,DEB
$,ARES,APRO,AALP,ABET
 IPGN=0
 RAD=57.29577951303232
```

```
      READ(5,1000) IPRO,ITR
1000  FORMAT(2I10)
      IPROB=IPRO
      WRITE(6,5000) IPRO,(MDNM(ITR,K),K=1,2)
      GO TO (10,100,200,200), ITR
10    ICOOR=2
      IPRINT=0
      ALNGTH=149598500.
      TM=86400.
      NTMC=2
      NDIM=6
      ACC=1.E-6
      DELTP=3.
      INPR=100
      TRTM1=0.
      ISP2=0
      NBOD=3
      IEPHEM=1
      IAUG=1
      READ(5,TRAJ)
15    CALL TIME(DATEJ,LYR,LMO,LDAY,LHR,LMIN,SECL,0)
      CALL TIME(FNDT ,IYR,IMO,IDAY,IHR,IMIN,SECI,0)
      FNTM=FNDT-DATEJ+TRTM1
      D=DATEJ+2415020.
      WRITE(6,5016) LMO,LDAY,LHR,LMIN,SECL,LYR,D
      D=FNDT+2415020.
      WRITE(6,5017) IMO,IDAY,IHR,IMIN,SECI,IYR,D
      WRITE(6,5018) TRTM1
      DELTM=FNTM
      WRITE(6,5038) IAUG
      WRITE(6,5013)
      IF(ICOOR.EQ.0) GO TO 16
      GO TO (17,18,19) ICOOR
16    WRITE(6,5031) (XI(I),I=1,NDIM)
      GO TO 31
17    WRITE(6,5032) (XI(I),I=1,NDIM)
      GO TO 31
18    WRITE(6,5033) (XI(I),I=1,NDIM)
      GO TO 31
19    WRITE(6,5034) RDS,PHI,THETA,VEL,GAMMA,SIGMA
31    IF(ICOOR-3) 20,30,30
20    IF(ICOOR-1) 50,40,40
30    PHI=PHI/RAD
      THETA=THETA/RAD
      GAMMA=GAMMA/RAD
      SIGMA=SIGMA/RAD
      CALL CONVERT(RDS,PHI,THETA,VEL,GAMMA,SIGMA,XI(1),XI(2),XI(3),XI(4)
     $,XI(5),XI(6))
40    NO(1)=4
      D=DATEJ
      CALL ORB(4,D)
      CALL EPHEM(1,D,1)
      VUNIT=ALNGTH/TM
      XP(1)=XP(1)*ALNGTH
      XP(2)=XP(2)*ALNGTH
      XP(3)=XP(3)*ALNGTH
      XP(4)=XP(4)*VUNIT
      XP(5)=XP(5)*VUNIT
      XP(6)=XP(6)*VUNIT
      T=D/36525.
```

```
      EPS=.4093197551-T*(.277111E-4+T*(.28604007E-7-T*.87751277E-8))
      CALL TRANS(2,XI(1),XI(2),XI(3),XI(4),XI(5),XI(6),XP(1),XP(2),
     $XP(3),XP(4),XP(5),XP(6),EPS,ICOOR)
      WRITE(6,5013)
      WRITE(6,5031) (XI(I),I=1,NDIM)
50    WRITE(6,5002) NTMC
      WRITE(6,5021)
      DO 51 I=1,NBOD
      J=NB(I)
51    WRITE(6,5051) PLANET(J)
      NTP=NB(3)
      WRITE(6,5022) PLANET(NTP)
      WRITE(6,5023) ALNGTH,TM
      TRTMB=TRTM1
      INCMT=0
      TIMINT=0.
      ISPH=0
      ICL=0
      IF(IEPHEM.NE.0) GO TO 80
      WRITE(6,2006)
2006  FORMAT(/////1X,130(1H*)///55X*MEAN ORBITAL ELEMENTS*///3X*PLANET*1
     $1X,*A*16X*E*16X*I*14X*NODE*13X*OMEGA*12X*PERI*12X*DATE*//)
      DO 60 I=1,11
60    DATE(I)=FNDT
      J=NB(2)
      DATE(J)=DATEJ
      DATE(11)=DATE(4)
      IF(MERCURY(1).EQ.0) GO TO 61
      SEC=MERCURY(5)
      CALL TIME(DATE(2),MERCURY(6),MERCURY(1),MERCURY(2),MERCURY(3),
     $MERCURY(4),SEC,0)
61    IF(VENUS(1)   .EQ.0) GO TO 62
      SEC=VENUS(5)
      CALL TIME(DATE(3),VENUS(6),VENUS(1),VENUS(2),VENUS(3),VENUS(4),SEC
     $,0)
62    IF(EARTH(1)   .EQ.0) GO TO 63
      SEC=EARTH(5)
      CALL TIME(DATE(4),EARTH(6),EARTH(1),EARTH(2),EARTH(3),EARTH(4),
     $SEC,0)
63    IF(MARS(1)    .EQ.0) GO TO 64
      SEC=MARS(5)
      CALL TIME(DATE(5),MARS(6),MARS(1),MARS(2),MARS(3),MARS(4),SEC,0)
64    IF(JUPITER(1).EQ.0) GO TO 65
      SEC=JUPITER(5)
      CALL TIME(DATE(6),JUPITER(6),JUPITER(1),JUPITER(2),JUPITER(3),
     $JUPITER(4),SEC,0)
65    IF(SATURN(1)  .EQ.0) GO TO 66
      SEC=SATURN(5)
      CALL TIME(DATE(7),SATURN(6),SATURN(1),SATURN(2),SATURN(3),
     $SATURN(4),SEC,0)
66    IF(URANUS(1)  .EQ.0) GO TO 67
      SEC=URANUS(5)
      CALL TIME(DATE(8),URANUS(6),URANUS(1),URANUS(2),URANUS(3),
     $URANUS(4),SEC,0)
67    IF(NEPTUNE(1).EQ.0) GO TO 68
      SEC=NEPTUNE(5)
      CALL TIME(DATE(9),NEPTUNE(6),NEPTUNE(1),NEPTUNE(2),NEPTUNE(3),
     $NEPTUNE(4),SEC,0)
68    IF(PLUTO(1)   .EQ.0) GO TO 69
      SEC=PLUTO(5)
```

```
      CALL TIME(DATE(10),PLUTO(6),PLUTO(1),PLUTO(2),PLUTO(3),PLUTO(4),
     $SEC,0)
69    IF(MOON(1)    .EQ.0) GO TO 74
      SEC=MOON(5)
      CALL TIME(DATE(11),MOON(6),MOON(1),MOON(2),MOON(3),MOON(4),SEC,0)
74    DO 70 I=1,NBOD
      J=NB(I)
      CALL ORB(J,DATE(J))
      CALL TIME (DATE(J),IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      GO TO (70,71,71,71,71,71,71,71,71,71,72),J
71    IL=8*(J-2)
      WRITE(6,1006)PLANET( J),ELMNT(IL+6),ELMNT(IL+4),ELMNT(IL+1),
     *ELMNT(IL+2),ELMNT(IL+3),ELMNT(IL+7),IMO,IDAY,IHR,IMIN,SECI,IYR
1006  FORMAT(1XA10,2X6(E15.8,1X),4I3,F7.3,I5)
      GO TO 70
72    WRITE(6,1005) PLANET(11),ELMNT(78),ELMNT(76),ELMNT(73),ELMNT(74),
     *ELMNT(75),IMO,IDAY,IHR,IMIN,SECI,IYR
1005  FORMAT(1XA10,2X5(E15.8,1X),16X,4I3,F7.3,I5)
70    CONTINUE
      WRITE(6,1008)
1008  FORMAT(//1X,130(1H*)//)
      GO TO 90
80    WRITE(6,1007)
1007  FORMAT(/8X*ORBITAL ELEMENTS WILL BE CALCULATED AT EVERY TIME INTER
     $VAL*)
90    IF(NTMC.NE.2) GO TO 95
      IF(IPRINT.NE.0) GO TO 91
      WRITE(6,5039)
      GO TO 92
91    WRITE(6,5040)
92    IF(ISP2.NE.0) GO TO 93
      WRITE(6,5041)
      GO TO 94
93    WRITE(6,5042)
94    WRITE(6,5043) ACC
      WRITE(6,5044) DELTP,INPR
      IPG=0
      DO 96 I=1,6
      RCA1(I)=0.
      RCA2(I)=0.
96    RCA3(I)=0.
      DO 97 I=1,3
      RSOI1(I)=0.
      RSOI2(I)=0.
      RSOI3(I)=0.
      VSOI1(I)=0.
      VSOI2(I)=0.
97    VSOI3(I)=0.
      ISOI1=0
      ISOI2=0
      ISOI3=0
      ICA1=0
      ICA2=0
      ICA3=0
      BSI1=0.
      BSI2=0.
      BSI3=0.
      BDTSI1=0.
      BDTSI2=0.
      BDTSI3=0.
```

```
      BDRSI1=0.
      BDRSI2=0.
      BDRSI3=0.
      TSOI1=1.E50
      TSOI2=1.E50
      TSOI3=1.E50
      TCA1=1.E50
      TCA2=1.E50
      TCA3=1.E50
95    GO TO (500,100,210,210), ITR
100   GO TO 500
200   ICOOR=2
      IPRINT=1
      ALNGTH=149598500.
      TM=86400.
      NTMC=2
      ACC=1.E-6
      DELTP=1.E50
      INPR=100000
      TRTM1=0.
      IAUG=1
      NENT=0
      NEV1=0
      NEV2=0
      NEV3=0
      NEV4=0
      ISTMC=1
      IDNF=0
      IMNF=0
      NST=3
      IEIG=1
      IHYP1=2
      ISP2=0
      NBOD=3
      IEPHEM=1
      NDIM=6
      FOP=1.E-15
      FOV=1.E-25
      U1=-.061351
      V1=.237886
      W1=-.969355
      U2=.028986
      V2=.960388
      W2=-.277141
      U3=.201963
      V3=.831343
      W3=-.517784
      DO 201 I=1,17
      DO 201 J=1,17
201   P(I,J)=0.
      DO 202 I=1,3
      P(I,I)=1.
202   P(I+3,I+3)=1.E-4
      SAL(1)=1.031
      SAL(2)=.05
      SAL(3)=.05
      SLAT(1)=35.384
      SLAT(2)=40.417
      SLAT(3)=-35.311
      SLON(1)=-116.833
```

```
      SLON(2)=-3.667
      SLON(3)=149.136
      DO 203 I=1,20
      ICDQ3(I)=1
203   ICDT3(I)=3
      MNCN(1)=1.E-6
      MNCN(2)=1.E-12
      MNCN(3)=1.E-6
      MNCN(4)=1.E-12
      MNCN(5)=1.E-6
      MNCN(6)=1.E-12
      MNCN(7)=1.E-6
      MNCN(8)=1.E-12
      MNCN(9)=2.5E-9
      MNCN(10)=2.5E-9
      MNCN(11)=2.5E-9
      MNCN(12)=2.5E-9
      FACP=1.
      FACV=1.E-4
      DO 205 I=1,50
205   TEV(I)=0.
      SIGRES=4.E-8
      SIGPRO=.0001
      SIGALP=.0043625
      SIGBET=.0043625
      ISTM1=0
      DTMAX=8.
      NDACC=0
      ACCND=2.5E-5
      DELAXS=100.
      DELECC=1.E-5
      DELICL=2.7777777E-3/RAD
      DELMUS=1.E7
      DELMUP=.1
      DMUSB=0.
      DMUPB=0.
      DAB=0.
      DEB=0.
      DIB=0.
      TTIM1=1.E50
      TTIM2=1.E50
      DO 204 I=1,3
      DO 204 J=1,3
204   UNMAC(I,J)=0.
      DO 208 I=1,9
208   SLB(I)=0.
      DO 209 I=1,12
209   BIA(I)=0.
      IAMNF=0
      NBOD1=11
      DO 218 I=1,11
218   NB1(I)=I
      ACC1=1.E-6
      DO 222    I=1,17
      XI(I)=0.
      XI1(I)=0.
      ZI(I)=0.
222   ADEVX(I)=0.
      DO 219 I=1,20
      ARES(I)=0.
```

```
      APRO(I)=0.
      AALP(I)=0.
219   ABET(I)=0.
      I=ITR-2
      GO TO(206,207),I
206   READ(5,ERRAN)
      GO TO 15
207   READ(5,SMLTN)
      GO TO 15
210   GO TO (211,213,212,215,213,216,214,216,214,216,217),IAUG
211   NDIM=6
      GO TO 220
212   NDIM=8
      GO TO 220
213   NDIM=9
      GO TO 220
214   NDIM=11
      GO TO 220
215   NDIM=12
      GO TO 220
216   NDIM=15
      GO TO 220
217   NDIM=17
220   WRITE(6,5011)
      IF(NENT) 230,221,230
221   TMN(1)=FNTM+1.
      WRITE(6,5035)
      NMN=1
      GO TO 247
230   READ(5,1550) ((SCHED(I,J),J=1,3),MEAS(I),I=1,NENT)
      WRITE(6,5003)
      WRITE(6,5004) ((SCHED(I,J),J=1,3),MEAS(I),I=1,NENT)
      DO 235 I=1,NENT
      AP(I)=SCHED(I,1)
      IF(AP(I)) 234,234,235
234   AP(I)=AP(I)+SCHED(I,3)
235   CONTINUE
      ICNT=0
      K=0
240   IROW=1
      AMIN=AP(1)
      IF(NENT.EQ.1) GO TO 246
      DO 245 I=2,NENT
      IF(AMIN.LE.AP(I)) GO TO 245
      AMIN=AP(I)
      IROW=I
245   CONTINUE
246   K=K+1
      TMN(K)=AMIN
      MCODE(K)=MEAS(IROW)
      AP(IROW) = AMIN+SCHED(IROW,3)
      IF(AP(IROW).LE.SCHED(IROW,2)) GO TO 240
      AP(IROW)=1.E50
      ICNT=ICNT+1
      IF(ICNT.LT.NENT) GO TO 240
      NMN=K
      DO 248 I=1,NMN
      IF(TMN(I).GT.FNTM) GO TO 249
248   CONTINUE
      GO TO 247
```

```
249   NMN=I-1
      WRITE(6,5047) NMN
247   IF(NEV1.EQ.0) T1(1)=1.E50
      IF(NEV2.EQ.0) T2(1)=1.E50
      IF(NEV3.EQ.0) T3(1)=1.E50
      IF(NEV4.EQ.0) T4(1)=1.E50
      NEV=NEV1+NEV2+NEV3+NEV4
      IF(NEV.NE.0) GO TO 250
      TEV(1)=FNTM+1.
      WRITE(6,5036)
      GO TO 33
250   AP(1)=T1(1)
      I1=1
      AP(2)=T2(1)
      I2=1
      AP(3)=T3(1)
      I3=1
      AP(4)=T4(1)
      I4=1
      ICNT=0
      K=0
251   AMIN=AP(1)
      IROW=1
      DO 255 I=2,4
      IF(AMIN.LE.AP(I)) GO TO 255
      AMIN=AP(I)
      IROW=I
255   CONTINUE
      K=K+1
      TEV(K)=AMIN
      IEVNT(K)=IROW
      GO TO (256,257,258,259), IROW
256   I1=I1+1
      AP(1)=T1(I1)
      IF(I1.LE.NEV1) GO TO 251
      AP(1)=1.E50
      ICNT=ICNT+1
      GO TO 260
257   I2=I2+1
      AP(2)=T2(I2)
      IF(I2.LE.NEV2) GOTO 251
      AP(2)=1.E50
      ICNT=ICNT+1
      GO TO 260
258   I3 = I3+1
      AP(3)=T3(I3)
      IF(I3.LE.NEV3) GO TO 251
      AP(3)=1.E50
      ICNT=ICNT+1
      GO TO 260
259   I4=I4+1
      AP(4)=T4(I4)
      IF(I4.LE.NEV4) GO TO 251
      AP(4)=1.E50
      ICNT=ICNT+1
260   IF(ICNT.LT.4) GO TO 251
      WRITE(6,5005)
      N1=0
      N2=0
      N3=0
```

```
      N4=0
      DO 261 I=1,NEV
      J=IEVNT(I)
      GO TO (262,263,264,265),
  262 N1=N1+1
      WRITE(6,5006) TEV(I),EVNM(J)
      GO TO 261
  263 N2=N2+1
      WRITE(6,5045) TEV(I),EVNM(J),TPT2(N2)
      GO TO 261
  264 N3=N3+1
      WRITE(6,5046) TEV(I),EVNM(J),ICDT3(N3),ICDQ3(N3)
      GO TO 261
  265 N4=N4+1
      WRITE(6,5006) TEV(I),EVNM(J)
  261 CONTINUE
      NGE=0
      NPE=0
      NQE=0
      DO 266 I=1,NEV
      IF(TEV(I).GT.FNTM) GO TO 267
  266 CONTINUE
      GO TO 268
  267 NEV=I-1
      WRITE(6,5048) NEV
  268 TEV(NEV+1)=FNTM+1.
      IPOL=0
      IIPOL=0
      IF(NEV1.EQ.0) GO TO 33
      WRITE(6,5007)
      GO TO(23,24,25),IHYP1
   23 WRITE(6,5008)
      GO TO 26
   24 WRITE(6,5009)
      GO TO 26
   25 WRITE(6,5010)
   26 CONTINUE
   33 WRITE(6,5015)
      DO 27 I=1,NDIM
   27 WRITE(6,5014) (P(I,J),J=1,I)
      WRITE(6,5011)
      WRITE(6,5028) ISTMC
      IF(ISTMC.NE.0) GO TO 34
      WRITE(6,5027) FACP,FACV
   34 IF(IDNF) 36,35,36
   35 WRITE(6,5024)
      GO TO 37
   36 WRITE(6,5025) DNCN
   37 IF (IMNF) 41,43,41
   43 WRITE(6,5029) ((MNNAME(I,J),J=1,3),MNCN(I),I=1,12)
      GO TO 42
   41 WRITE(6,5030)
   42 IF(NST.EQ.0) GO TO 280
      WRITE(6,5019)
      WRITE(6,5020) (I,SAL(I),SLAT(I),SLON(I),I=1,NST)
  270 DO 271 I=1,NST
      SAL(I)=SAL(I)/RAD
  271 SLON(I)=SLON(I)/RAD
  280 CALL GHA
      OMEGA=6.300387432
```

```
      DO 281 I=1,NDIM
      DO 281 J=1,NDIM
281   PG(I,J)=P(I,J)
      DO 282 I=1,6
282   XG(I)=XI(I)
      TG=TRTM1
      MCNTR=1
      DO 283 I=1,NDIM
      XB(I)=XI(I)
      XF(I)=XI(I)
      DO 283 J=1,NDIM
283   PB(I,J)=P(I,J)
      D=DATEJ-2415020.
      T=D/36525.
      EPS=.4093197551-T*(.277111E-4+T*(.28604007E-7-T*.87751277E-8))
      DO 284 I=1,2
      DO 284 J=1,6
284   EM(I,J)=0.
      I=ITR-2
      GO TO (500,300),I
300   DO 310 I=1,NDIM
310   EDEVX(I)=0.
      DO 320 I=1,NDIM
      XI1(I)=XI(I)
      ADEVXB(I)=ADEVX(I)
320    Z(I)=XI(I)+ADEVX(I)
      WRITE(6,5049) (ADEVX(I),I=1,NDIM)
      WRITE(6,5050)
      DO 330 I=1,NBOD1
      J=NB1(I)
330   WRITE(6,5051) PLANET(J)
      WRITE(6,5052) ACC1
      WRITE(6,5053) ((MNNAME(I,J),J=1,3),BIA(I),I=1,12)
      WRITE(6,5011)
      WRITE(6,5054) DMUSB,DMUPB,DAB,DEB,DIB
      WRITE(6,5055)
      IF(FNTM.LE.TTIM1) GO TO 350
      IF(FNTM.LE.TTIM2) GO TO 340
      WRITE(6,5056) TRTM1,TTIM1,(UNMAC(I,1),I=1,3)
      WRITE(6,5056) TTIM1,TTIM2,(UNMAC(I,2),I=1,3)
      WRITE(6,5056) TTIM2,FNTM ,(UNMAC(I,3),I=1,3)
      GO TO 360
340   WRITE(6,5056) TRTM1,TTIM1,(UNMAC(I,1),I=1,3)
      WRITE(6,5056) TTIM1,FNTM ,(UNMAC(I,2),I=1,3)
      GO TO 360
350   WRITE(6,5056) TRTM1,FNTM ,(UNMAC(I,1),I=1,3)
360   WRITE(6,5057) (SLB(I),I=1,9)
      SLB(2)=SLB(2)/RAD
      SLB(3)=SLB(3)/RAD
      SLB(5)=SLB(5)/RAD
      SLB(6)=SLB(6)/RAD
      SLB(8)=SLB(8)/RAD
      SLB(9)=SLB(9)/RAD
      IF(IAMNF.GT.0) GO TO 370
      WRITE(6,5058)
      GO TO 380
370   WRITE(6,5059) ((MNNAME(I,J),J=1,3),AVARM(I),I=1,12)
380   CONTINUE
500   RETURN
1550  FORMAT(3F10.0 ,I10)
```

```
5000  FORMAT(1H1,4X*I N P U T   D A T A   F O R   P R O B L E M . . . .*
     $I5////7X*MODE TO BE EXECUTED. . .*2A10)
5016  FORMAT(/8X*LAUNCH DATE* 10X,4I3,F7.3,I5,5X*JULIAN DATE . . . *
     $F16.8)
5017  FORMAT(/8X*FINAL DATE*11X,4I3,F7.3,I5,5X*JULIAN DATE . . . *F16.8)
5018  FORMAT(/8X*INITIAL TRAJECTORY TIME =*F10.4)
5002  FORMAT(/8X*NOMINAL TRAJECTORY CODE. . .*I2)
5003  FORMAT(/8X*MEASUREMENT SCHEDULE*)
5004  FORMAT(10X*FROM*F8.2* DAYS TO*F8.2* DAYS, EVERY*F8.2* DAYS, MEASUR
     $E CODE*I5)
5005  FORMAT(/8X*EVENT SCHEDULE*/10X*TIME OF EVENT*10X*EVENT*10X
     $EVENT INFORMATION*)
5006  FORMAT(13X,F8.3,10X,A10)
5007  FORMAT(/10X*FOR EIGENVALUE EVENTS, THE SIGMA LEVEL OF THE HYPERELL
     $IPSOID IS*)
5008  FORMAT(1H+,73X,*K = 1*)
5009  FORMAT(1H+,73X,*K = 3*)
5010  FORMAT(1H+,73X,*K =1 AND K = 3*)
5011  FORMAT(1H1)
5013  FORMAT(/8X*INITIAL STATE VECTOR*)
5014  FORMAT(10X,6E20.8)
5015  FORMAT(/8X*INITIAL COVARIANCE MATRIX*)
5019  FORMAT(/8X*STATION LOCATION CONSTANTS*)
5020  FORMAT(10X*STATION NO. *I1,5X*ALTITUDE = *E15.8,5X*LATITUDE = *
     $E15.8,5X,*LONGITUDE = *E15.8)
5021  FORMAT(/8X*NOMINAL TRAJECTORY INFORMATION*//10X*BODIES TO BE CONSI
     $DERED*)
5022  FORMAT(/10X*TARGET PLANET. . .*A10)
5023  FORMAT(/8X*UNITS*/10X E15.8*/A.U.*20X,E15.8*/DAY*)
5024  FORMAT(/8X*DYNAMIC NOISE IS ZERO*)
5025  FORMAT(/8X*DYNAMIC NOISE CONSTANTS*/10X6E20.12)
5027  FORMAT(10X*NUMERICAL DIFFERENCING INFORMATION*/10X*POSITION FACTOR
     $ *E15.8/10X*VELOCITY FACTOR *E15.8)
5028  FORMAT(/8X*STATE TRANSITION MATRIX CODE . . .* I2)
5029  FORMAT(/8X*MEASUREMENT NOISE IS CONSTANT*12(/10X,3A10,E20.12))
5030  FORMAT(/8X*MEASUREMENT NOISE IS TO BE CALCULATED*)
5031  FORMAT(10X*HELIOCENTRIC ECLIPTIC COORDINATES*/(10XE20.8))
5032  FORMAT(10X*GEOCENTRIC EQUATORIAL COORDINATES*/(10XE20.8))
5033  FORMAT(10X*GEOCENTRIC ECLIPTIC COORDINATES  */(10XE20.8))
5034  FORMAT(23X*RDS*17X*PHI*16X*THETA*16X*VEL*16X*GAMMA*15X*SIGMA*/
     $10X6E20.8)
5035  FORMAT(/8X*NO MEASUREMENTS*)
5036  FORMAT(/8X*NO EVENTS*)
5037  FORMAT(/8X*EPHEMERIS IS TO BE UPDATED EVERY*F10.3*DAYS*)
5038  FORMAT(/8X*AUGMENTATION CODE. . . . .*I3)
5039  FORMAT(/8X*OUTPUT FROM VIRTUAL MASS PROGRAM WILL BE PRINTED AS USU
     $AL*)
5040  FORMAT(/8X *OUTPUT FROM VIRTUAL MASS PROGRAM WILL BE SUPPRESSED AT
     $ INITIAL AND FINAL STEPS*)
5041  FORMAT(/8X*VIRTUAL MASS PROGRAM WILL INTEGRATE UNTIL REACHING A NO
     $RMAL STOPPING CONDITION*)
5042  FORMAT(/8X*VIRTUAL MASS PROGRAM WILL STOP INTEGRATING UPON REACHIN
     $G SPHERE OF INFLUENCE OF TARGET PLANET*)
5043  FORMAT(/8X*ACCURACY FIGURE. . . . .*E12.5)
5044  FORMAT(/8X*PRINT INTERVALS*/10XE12.5,* DAYS*/10XI10* INCREMENTS*)
5045  FORMAT(13X,F8.3,10X,A10,10X*PREDICTING TO TIME*F7.2)
5046  FORMAT(13X,F8.3,10X,A10,10X*GUIDANCE POLICY*I5*, EXECUTION ERROR C
     $ODE*I5)
5047  FORMAT(/8X*NOTE--ONLY THE FIRST *I5* MEASUREMENTS WILL BE INCLUDED
     $ IN THE ANALYSIS SINCE THE OTHERS DO NOT OCCUR UNTIL AFTER THE FIN
```

```
     $AL TIME*)
5048  FORMAT(/8X*NOTE--ONLY THE FIRST *I5* EVENTS WILL BE INCLUDED IN TH
     $E ANALYSIS SINCE THE OTHERS DO NOT OCCUR UNTIL AFTER THE FINAL TIM
     $E*)
5049  FORMAT(/8X*ACTUAL DEVIATION OF STATE VECTOR AT INITIAL TIME*/
     $(10X,E20.8))
5050  FORMAT(/8X*ACTUAL TRAJECTORY INFORMATION*//10X*BODIES TO BE CONSID
     $ERED*)
5051  FORMAT(12X,A10)
5052  FORMAT(/8X*ACCURACY FIGURE FOR ACTUAL TRAJECTORY. .*E12.5)
5053  FORMAT(/8X*ACTUAL MEASUREMENT BIASES*12(/10X3A10,E20.12))
5054  FORMAT(/8X*DYNAMIC CONSTANT BIASES TO BE USED IN THE DETERMINATION
     $ OF THE ACTUAL TRAJECTORY*/10X*GRAVITATIONAL CONSTANT OF SUN. . .
     $. . . .*E20.13/10X*GRAVITATIONAL CONSTANT OF TARGET PLANET. .*E20.
     $13/10X*SEMI-MAJOR AXIS OF TARGET PLANET . . . . .*E20.13/
     $    10X*ECCENTRICITY OF TARGET PLANET. . . . . . .*E20.13/
     $    10X*INCLINATION OF TARGET PLANET . . . . . . .*E20.13)
5055  FORMAT(/8X*ACTUAL UNMODELLED ACCELERATION TO BE USED TO CALCULATE
     $THE ACTUAL DYNAMIC NOISE BY THE FOLLOWING SCHEDULE*//69X*X*24X*Y*
     $24X*Z*)
5056  FORMAT(10X*FROM*F8.3* DAYS THROUGH *F8.3* DAYS. . .*3E25.13)
5057  FORMAT(/8X*BIASES IN LOCATIONS OF ROTATING STATIONS*//22X*ALTITUDE
     $*17X*LATITUDE*17X*LONGITUDE*/10X*1*3E25.13/10X*2*3E25.13/10X*3*
     $3E25.13)
5058  FORMAT(/8X*THE UNCERTAINTIES IN THE ACTUAL MEASUREMENT NOISE ARE A
     $SSUMED TO BE*/8X*THE SAME AS THE UNCERTAINTIES IN THE MEASUREMENT
     $NOISE OF THE MOST RECENT NOMINAL*)
5059  FORMAT(/8X*THE ACTUAL MEASUREMENT NOISE WILL BE CALCULATED FROM TH
     $E FOLLOWING CONSTANTS*12(/10X3A10,E20.13))
      END
```

```
      SUBROUTINE DYNO(ICODE)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DLETP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      IF(ICODE.NE.0) GO TO 30
      DO 5 I=1,NDIM
      DO 5 J=1,NDIM
 5    Q(I,J)=0.
      IF(IDNF) 20,20,6
 6    D2=DELTM*TM*DELTM*TM
      DO 10 I=1,3
      Q(I,I)=DNCN(I)*D2*D2*.25
 10   Q(I+3,I+3)=DNCN(I)*D2
 20   RETURN
 30   DO 31 I=1,NDIM
 31   W(I)=0.
      T1=TRTM1
      T2=TRTM1+DELTM
      IF(T1.GT.TTIM1) GO TO 70
      IF(T2.GT.TTIM1) GO TO 50
      I=1
      IC=1
      DT=DELTM*TM
 40   D2=DT*DT
      W(1)=.5*UNMAC(1,I)*D2+W(1)
      W(2)=.5*UNMAC(2,I)*D2+W(2)
      W(3)=.5*UNMAC(3,I)*D2+W(3)
      W(4)=UNMAC(1,I)*DT+W(4)
      W(5)=UNMAC(2,I)*DT+W(5)
      W(6)=UNMAC(3,I)*DT+W(6)
      GO TO (20,51,61,62,81), IC
 50   IF(T2.GT.TTIM2) GO TO 60
      DT=(T2-TTIM1)*TM
      I=2
      IC=2
      GO TO 40
 51   DT=(TTIM1-T1)*TM
      I=1
      IC=1
      GO TO 40
 60   DT=(T2-TTIM2)*TM
      I=3
      IC=3
      GO TO 40
 61   DT=(TTIM2-TTIM1)*TM
      I=2
      IC=4
      GO TO 40
 62   DT=(TTIM1-T1)*TM
      I=1
```

```
      IC=1
      GO TO 40
70    IF(T1.GT.TTIM2) GO TO 90
      IF(T2.GT.TTIM2) GO TO 80
      DT=DELTM*TM
      I=2
      IC=1
      GO TO 40
80    DT=(T2-TTIM2)*TM
      I=3
      IC=5
      GO TO 40
81    DT=(TTIM2-T1)*TM
      I=2
      IC=1
      GO TO 40
90    DT=DELTM*TM
      I=3
      IC=1
      GO TO 40
      END
```

```
      SUBROUTINE EIGEN(RI,TEVN)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR THE LOGIC USED IN AN EIGEN-
C     VECTOR EVENT.
C
C     EIGEN USES THE FOLLOWING SUBROUTINES
C        NTM
C        PSIM
C        DYNO
C        NAVM
C        JACOBI
C        HYELS
C
C
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION PEIG(3,3),EGVL(3),EGVCT(3,3),RI(6),RF(6)
      DIMENSION VEIG(9),RHO(17,17)
      MAX=60
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
13    DO 5 I=1,6
5     XF(I)=RF(I)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
      WRITE(6,3006) TEVN
      WRITE(6,3001) (CMPNM(IAUG,I),XF(I),I=1,NDIM)
      LINES=LINES+NDIM+1
      CALL PSIM(RI,RF,ISTMC)
      WRITE (6,3002) TEVN,TRTM1
      LINES=LINES+5
      DO 6 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 1
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
1     IF (NDIM.EQ.6) GO TO 4
      WRITE (6,3013) I
      LINES=LINES+1
4     WRITE (6,3014) (PSI(I,J),J=1,NDIM)
6     LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(0)
```

```
      IF (LINES.LT.MAX-8) GO TO 2
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
2     WRITE (6,3003)
      WRITE (6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF(LINES.LT.MAX-9) GO TO 3
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
3     WRITE(6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 7 I=1,NDIM
      IF(LINES.LT. MAX-4) GO TO 8
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
8     IF (NDIM.EQ.6) GO TO 9
      WRITE (6,3013)I
      LINES=LINES+1
9     WRITE (6,3014) (P(I,J),J=1,NDIM)
7     LINES=LINES+(NDIM-1)/6+1
      ICODE=0
      DO 10 I=1,3
      DO 10 J=1,3
10    PEIG(I,J)=P(I,J)
      K=0
      DO.98 J=1,3
      DO 98 I=1,3
      K=K+1
98    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF(LINES.LT.MAX-16) GO TO 11
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
11    WRITE (6,1000) (I,EGVL(I),I=1,3)
      WRITE(6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
15    IF(IHYP1-2) 20,30,20
20    IF(LINES.LT.MAX-16) GO TO 21
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
21    CALL HYELS(1,PEIG,3)
      LINES=LINES+16
30    IF(IHYP1-1) 50,50,40
40    IF(LINES.LT.MAX-16) GO TO 41
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
41    CALL HYELS(3,PEIG,3)
      LINES=LINES+16
50    IF(ICODE) 55,55,80
55    IF(IEIG) 80,80,60
60    DO 70 I=1,3
      DO 70 J=1,3
70    PEIG(I,J)=P(I+3,J+3)
```

```
      K=0
      DO 99 J=4,6
      DO 99 I=4,6
      K=K+1
99    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
      IF(LINES.LT.MAX-16) GO TO 71
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
71    WRITE (6,2000) (I,EGVL (I),I=1,3)
      WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      ICODE=1
      GO TO 15
80    TRTM1=TEVN
      DO 90 I=1,NDIM
90    XI(I)=XF(I)
      DO 100 I=1,NDIM
      DO 100 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
100   RHO(J,I)=RHO(I,J)
      IF(LINES.LT.MAX-9) GOTO 101
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
101   WRITE(6,3007) TEVN
3007  FORMAT(///8X*CORRELATION COEFFICIENT MATRIX AT TIME OF EIGENVECTOR
     $ EVENT  -- *F8.3* DAYS*/)
      LINES=LINES+5
      DO 104 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 102
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
102   IF(NDIM.EQ.6) GO TO 103
      WRITE(6,3013) I
      LINES=LINES+1
103   WRITE(6,3014) (RHO(I,J),J=1,NDIM)
104   LINES=LINES+(NDIM-1)/6+1
1000  FORMAT(///20X*POSITION EIGENVALUES*/3(22X,I2,E25.13/))
1001  FORMAT(///20X*POSITION EIGENVECTORS*/3(22XI2,3E25.13/))
2000  FORMAT(///20X*VELOCITY EIGENVALUES*/3(22X,I2,E25.13/))
2001  FORMAT (///20X*VELOCITY EIGENVECTORS*/3(22XI2,3E25.13/))
3000  FORMAT(1H1//8X2A10*--   EIGENVECTOR EVENT AT TRAJECTORY TIME *F12.
     $3* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*)//)
3001  FORMAT(10XA10,E20.13)
3002  FORMAT(///8X*STATE TRANSITION MATRIX  --  PSI(*F12.3*,*F12.3*)*/)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF EIGENVECTOR EVENT  --  P
     $(*F12.3*,*F12.3*)*/)
3006  FORMAT(8X*STATE VECTOR AT TIME *F12.3* DAYS*/)
3013  FORMAT (10X*ROW*I3)
3014  FORMAT(16X,6E17.8)
      RETURN
      END
```

```
      SUBROUTINE EIGSIM(RI,TEVN,RI1)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),RI1(6),RF1(6),RI2(6),RF2(6),DUM(17)
      DIMENSION PEIG(3,3),EGVL(3),EGVCT(3,3)
      DIMENSION VEIG(9),RHO(17,17)
      MAX=60
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
      DO 10 I=1,6
10    XF(I)=RF(I)
      IF(NQE.NE.0) GO  TO 20
      DO 11 I=1,NDIM
11    XF1(I)=XF(I)
      DO 12 I=1,6
12    RF1(I)=RF(I)
      GO TO 30
20    CALL NTM(RI1,RF1,NTMC,2)
      DO 21 I=1,6
21    XF1(I)=RF1(I)
30    CALL PSIM(RI1,RF1,ISTMC)
      CALL DYNO(0)
      CALL NAVM(1,1)
      DO 50 I=1,NDIM
      DO 50 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
50    RHO(J,I)=RHO(I,J)
      DO 39 I=1,6
39    RI2(I)=XI1(I)+ADEVX(I)
      CALL NTM(RI2,RF2,NTMC,3)
      DO 40 I=1,6
40    Z(I)=RF2(I)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
      WRITE(6,3006)
      LINES=LINES+3
      WRITE(6,3001) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
      LINES=LINES+NDIM
      WRITE(6,3002) TEVN,TRTM1
```

```
      LINES=LINES+5
      DO 33 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 31
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
31    IF(NDIM.EQ.6) GO TO 32
      WRITE(6,3013) I
      LINES=LINES+1
32    WRITE(6,3014) (PSI(I,J),J=1,NDIM)
33    LINES=LINES+(NDIM-1)/6+1
      IF(LINES.LT.MAX-8) GO TO 34
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
34    WRITE(6,3003)
      WRITE(6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      IF(LINES.LT.MAX-9) GOTO 35
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
35    WRITE(6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 38 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 36
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
36    IF(NDIM.EQ.6) GO TO 37
      WRITE(6,3013) I
      LINES=LINES+1
37    WRITE(6,3014) (P(I,J),J=1,NDIM)
38    LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(1)
      DO 43 I=1,6
43    ADEVX(I)=Z(I)+W(I)-XF1(I)
      DO 45 I=1,NDIM
      DUM(I)=0.
      DO 45 J=1,NDIM
45    DUM(I)=DUM(I)+PSI(I,J)*EDEVX(J)
      DO 46 I=1,NDIM
46    EDEVX(I)=DUM(I)
      ICODE=0
      DO 60 I=1,3
      DO 60 J=1,3
60    PEIG(I,J)=P(I,J)
      K=0
      DO 61 J=1,3
      DO 61 I=1,3
      K=K+1
61    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF(LINES.LT.MAX-16) GO TO 62
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
62    WRITE (6,1000) (I,EGVL(I),I=1,3)
      WRITE(6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
```

```
65      IF(IHYP1-2) 70,80,70
70      IF(LINES.LT.MAX-16) GO TO 71
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
71      CALL HYELS(1,PEIG,3)
        LINES=LINES+16
80      IF(IHYP1-1) 100,100,90
90      IF(LINES.LT.MAX-16) GO TO 91
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
91      CALL HYELS(3,PEIG,3)
        LINES=LINES+16
100     IF(ICODE) 105,105,130
105     IF(IEIG) 130,130,110
110     DO 120 I=1,3
        DO 120 J=1,3
120     PEIG(I,J)=P(I+3,J+3)
        K=0
        DO 149 J=4,6
        DO 149 I=4,6
        K=K+1
149     VEIG(K)=P(I,J)
        CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
        IF(LINES.LT.MAX-16) GO TO 121
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
121     WRITE (6,2000) (I,EGVL (I),I=1,3)
        WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
        LINES=LINES+16
        ICODE=1
        GO TO 65
130     IF(LINES.LT.MAX-9) GO TO 51
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
51      WRITE(6,3010)
        LINES=LINES+5
        DO 54 I=1,NDIM
        IF(LINES.LT.MAX-4) GO TO 52
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
52      IF(NDIM.EQ.6) GO TO 53
        WRITE(6,3013) I
        LINES=LINES+1
53      WRITE(6,3014) (RHO(I,J),J=1,NDIM)
54      LINES=LINES+(NDIM-1)/6+1
        IF(LINES.LT.MAX-NDIM-5) GO TO 42
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
42      WRITE(6,3007) (W(I),I=1,NDIM)
        LINES=LINES+NDIM+5
        IF(LINES.LT.MAX-NDIM-7) GO TO 47
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
        LINES=9
```

```
47    WRITE(6,3009)(EDEVX(I),ADEVX(I),I=1,NDIM)
      LINES=LINES+NDIM+7
      TRTM1=TEVN
      DO 140 I=1,NDIM
      XI1(I)=XF1(I)
140   XI(I)=XF(I)
1000  FORMAT(///20X*POSITION EIGENVALUES*/3(22X,I2,E25.13/))
1001  FORMAT(///20X*POSITION EIGENVECTORS*/3(22XI2,3E25.13/))
2000  FORMAT(///20X*VELOCITY EIGENVALUES*/3(22X,I2,E25.13/))
2001  FORMAT (///20X*VELOCITY EIGENVECTORS*/3(22XI2,3E25.13/))
3000  FORMAT(1H1*//8X2A10*--EIGENVECTOR EVENT AT TRAJECTORY TIME *F12.3
     $* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8 ///1X,130(1H*)//)
3001  FORMAT(8XA10,5X,E20.10,5X,E20.10,5X,E20.10)
3002  FORMAT(///8X*STATE TRANSITION MATRIX  --  PSI(*F8.3*,*F8.3*)*/)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF EIGENVALUE EVENT  --  P(
     $*F8.3*,*F8.3*)*/)
3006  FORMAT(8X*STATE VECTOR*//22X*ORIGINAL NOMINAL*7X*MOST RECENT NOMIN
     $AL*13X*ACTUAL*)
3007  FORMAT(///8X*ACTUAL DYNAMIC NOISE*//(8XE20.10))
3009  FORMAT(///8X*DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NO
     $MINAL TRAJECTORY*//15X*ESTIMATED*13X*ACTUAL*/(8X2E20.10))
3010  FORMAT(///8X*CORRELATION COEFFECIENT MATRIX*/)
3013  FORMAT(10X*ROW *I3)
3014  FORMAT(16X6E17.8)
      RETURN
      END
```

```
      SUBROUTINE EPHEM(IC,D,N)
C
C
C
C     THIS SUBROUTINE DETERMINES THE MEAN ANOMALY (TRUE ANOMALY FOR
C     MOON) AT A GIVEN TIME FOR EACH OF THE PLANETS AND USES THIS
C     INFORMATION TOGETHER WITH THE MEAN ORBITAL ELEMENTS IN -ORB- TO
C     CALCULATE THE HELIOCENTRIC ECLIPTIC RECTANGULAR COORDINATES OF
C     EACH OF THE PLANETS.
C
C     ARGUMENTS
C
C        IC       --  EPHEMERIS CODE
C                     =0  THE COORDINATES OF THE PLANET ARE RETURNED
C                         IN THE ARRAY F AS SHOWN BELOW
C                         F(4*NO(I)-3,1) = X
C                         F(4*NO(I)-3,2) = Y
C                         F(4*NO(I)-3,3) = Z
C                         F(4*NO(I)-2,1) = VX
C                         F(4*NO(I)-2,2) = VY
C                         F(4*NO(I)-2,3) = VZ
C                     =1  THE COORDINATES ARE RETURNED IN AN ARRAY XP
C        D        --  DATE AT WHICH COORDINATES ARE TO BE DETERMINED
C        N        --  NUMBER OF BODIES FOR WHICH COORDINATES ARE TO BE
C                     DETERMINED
C
C
C     WHERE I=1,NBODYI.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DIMENSION TRG(14)
      FN1(A,B,C,D,X)=A+B*X+C*X*X+D*X*X*X
      FN2(A,B,X)=A+B*X
      PI2=2.*PI
      T= D/36525.
      DD= D*1.E-4
      ITEST2=0
      DO 90IJ=1,N
      IJ1=IJ
      ITEST=0
      I=NO(IJ)-1
      IF(I)90,90,10
 10   GO TO (30,30,20,30,40,40,40,40,40,80)I
 20   IF(ITEST2)90,30,90
 30   IL=8*(I-1)
      K=20*(I-1)+17
      ELMNT(IL+5)=FN1(CN(K),CN(K+1),CN(K+2),CN(K+3),DD)
      GO TO 50
 40   K=10*(I-5)+9
      IL=8*(I-1)
      ELMNT(IL+5)=FN2(ST(K),ST(K+1),DD)
 50   ITEMP=ELMNT(IL+5)/PI2
```

```
      ELMNT(IL+5)=ELMNT(IL+5)-FLOAT(ITEMP)*PI2
      E2=ELMNT(IL+4)*ELMNT(IL+4)
      E3=E2*ELMNT(IL+4)
      ELMNT(IL+8)=ELMNT(IL+5)+(ELMNT(IL+4)-E3/8.)*SIN(ELMNT(IL+5))+.5*
     1E2*SIN(2.*ELMNT(IL+5))+.375*E3*SIN(3.*ELMNT(IL+5))
      P=ELMNT(IL+6)*(1.-ELMNT(IL+4)*ELMNT(IL+4))
      IND=4*IJ1-3
      TRG(1)=COS(ELMNT(IL+8))
      TRG(2)=SIN(ELMNT(IL+8))
      R=ELMNT(IL+6)*(1.-ELMNT(IL+4)*TRG(1))
      VEL=SQRT(PMASS(1)*(2./R-1./ELMNT(IL+6)))
      TRG(3)=(P-R)/(ELMNT(IL+4)*R)
      TRG(4)=SQRT(1.-TRG(3)*TRG(3))
      TRG(5)=COS(ELMNT(IL+1))
      TRG(6)=SIN(ELMNT(IL+1))
      TRG(7)=COS(ELMNT(IL+2))
      TRG(8)=SIN(ELMNT(IL+2))
      TRG(9)=COS(ELMNT(IL+7))
      TRG(10)=SIN(ELMNT(IL+7))
      TRG(11)=SQRT(PMASS(1)*P)/(R*VEL)
      TRG(12)=SQRT(1.-TRG(11)*TRG(11))
      IF(TRG(2))60,60,70
60    TRG(4)=-TRG(4)
      TRG(12)=-TRG(12)
70    TRG(13)=TRG(9)*TRG(3)-TRG(10)*TRG(4)
      TRG(14)=TRG(10)*TRG(3)+TRG(9)*TRG(4)
      IF(IC-1) 71,100,71
71    F(IND,1)=R       *(TRG(13)*TRG(7)-TRG(14)*TRG(8)*TRG(5))
      F(IND,2)=R           *(TRG(13)*TRG(8)+TRG(14)*TRG(7)*TRG(5))
      F(IND,3)=R           *TRG(14)*TRG(6)
      WX=TRG(6)*TRG(8)
      WY=-TRG(6)*TRG(7)
      WZ=TRG(5)
      FCTR=VEL/R
      F(IND+1,1)=FCTR*((WY*F(IND,3)-WZ*F(IND,2))*TRG(11)+F(IND,1)*TRG(12
     1))
      F(IND+1,2)=FCTR*((WZ*F(IND,1)-WX*F(IND,3))*TRG(11)+F(IND,2)*TRG(12
     1))
      F(IND+1,3)=FCTR*((WX*F(IND,2)-WY*F(IND,1))*TRG(11)+F(IND,3)*TRG(12
     1))
      IF(ITEST)79,90,79
79    IK=IJ1
      GO TO 86
80    ELMNT(80)=FN1(EMN(9),EMN(10)*36525.,EMN(11),EMN(12),T)
      ITEMP=ELMNT(80)/PI2
      ELMNT(80)=ELMNT(80)-FLOAT(ITEMP)*PI2
      ELMNT(77)=ELMNT(80)-ELMNT(75)
      IF(ELMNT(77).LT.0.)ELMNT(77)=ELMNT(77)+PI2
      ELMNT(79)=ELMNT(80)-ELMNT(74)
      TRG(3)=COS(ELMNT(77))
      TRG(4)=SIN(ELMNT(77))
      TRG(5)=COS(ELMNT(73))
      TRG(6)=SIN(ELMNT(73))
      TRG(7)=COS(ELMNT(74))
      TRG(8)=SIN(ELMNT(74))
      P=ELMNT(78)*(1.-ELMNT(76)*ELMNT(76))
      R      =P/(ELMNT(76)*TRG(3)+1.)
      VEL=SQRT(PMASS(4)*(2./R-1./ELMNT(78)))
      TRG(11)=SQRT(PMASS( 4)*P)/(R*VEL)
      TRG(12)=SQRT(1.-TRG(11)*TRG(11))
```

```
      IF(TRG(4).LT.0.)TRG(12)=-TRG(12)
      TRG(13)=COS(ELMNT(79))
      TRG(14)=SIN(ELMNT(79))
      IN=4*IJ-3
      F(IN,1)=R      *(TRG(13)*TRG(7)-TRG(14)*TRG(8)*TRG(5))
      F(IN,2)=R      *(TRG(13)*TRG(8)+TRG(14)*TRG(7)*TRG(5))
      F(IN,3)=R      *TRG(14)*TRG(6)
      WX=TRG(6)*TRG(8)
      WY = -TRG(6)*TRG(7)
      WZ=TRG(5)
      FCTR=VEL/R
      F(IN+1,1)=FCTR*((WY*F(IN,3)-WZ*F(IN,2))*TRG(11)+F(IN,1)*TRG(12))
      F(IN+1,2)=FCTR*((WZ*F(IN,1)-WX*F(IN,3))*TRG(11)+F(IN,2)*TRG(12))
      F(IN+1,3)=FCTR*((WX*F(IN,2)-WY*F(IN,1))*TRG(11)+F(IN,3)*TRG(12))
      DO 81 IK=1,NBODYI
      IF(NO(IK).EQ.4)GO TO 84
81    CONTINUE
      IJ1=NBODYI+1
      GO TO 85
84    IF(IK.LT.IJ)GO TO 86
      IJ1=IK
      ITEST2=1
85    ITEST=1
      I=3
      GO TO 30
86    IK=4*IK-3
      DO 87 J=1,3
      F(IN,J)=F(IN,J)+F(IK,J)
87    F(IN+1,J)=F(IN+1,J)+F(IK+1,J)
90    CONTINUE
      RETURN
100   XP(1)=   R      *(TRG(13)*TRG(7)-TRG(14)*TRG(8)*TRG(5))
      XP(2)=   R         *(TRG(13)*TRG(8)+TRG(14)*TRG(7)*TRG(5))
      XP(3)=   R         *TRG(14)*TRG(6)
      WX=TRG(6)*TRG(8)
      WY=-TRG(6)*TRG(7)
      WZ=TRG(5)
      FCTR=VEL/R
      XP(4)=FCTR*((WY*XP(3)-WZ*XP(2))*TRG(11)+XP(1)*TRG(12))
      XP(5)=FCTR*((WZ*XP(1)-WX*XP(3))*TRG(11)+XP(2)*TRG(12))
      XP(6)=FCTR*((WX*XP(2)-WY*XP(1))*TRG(11)+XP(3)*TRG(12))
      RETURN
      END
```

```
      SUBROUTINE ESTMT(D1,DELTM,TRTM)
C
C
C
C     THIS SUBROUTINE UPDATES THE FINAL VALUES OF PRECEDING COMPUTING
C     INTERVAL TO SERVE AS INITIAL VALUES FOR THE NEW STEP, DETERMINES
C     THE DESIRED SIZE OF TIME INCREMENT ON THE BASIS OF TRUE ANOMALY OR
C     REQUESTED PRINT TIME, AND ESTIMATES THE FINAL POSITION AND
C     MAGNITUDE OF THE VIRTUAL MASS USING FORMULAS (III-16) IN
C
C          NOVAK, D. H. -VIRTUAL MASS TECHNIQUE FOR COMPUTING SPACE
C          TRAJECTORIES-, FINAL REPORT, CONTRACT NO. NAS 9-4370,
C          ER 14045, MARTIN, BALTIMORE DIVISION, JANUARY, 1966.  PG. 23.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
C INDEX VALUES IN V ARRAY
      INCMNT=INCMNT+1
      DO 361 I=1,9,2
      DO 360 J=1,4
  360 V(I,J)=V(I+1,J)
  361 CONTINUE
      IF(INC.EQ.0) GO TO 370
      IF(INCMNT.EQ.INCMNT/INCPR*INCPR) KOUNT =1
370   ITRAT=1
C ESTABLISH COMPUTING TIME INCREMENT
      V(11,1)=SQRT(V(11,2)*V(11,2)+V(11,3)*V(11,3)+V(11,4)*V(11,4))
      V(7,6)=V(3,6)*V(9,1)/V(11,1)
      IF(V(2,5).GT.V(1,1)+V(7,6)) GO TO 365
      V(7,6)=V(2,5)-V(1,1)
      V(2,1)=V(2,5)
      V(4,1)=D1+DELTM
      KOUNT=1
      GO TO 391
365   IF (IPR.EQ.0) GO TO 378
  394 IF(V(1,1)+1.1*V(7,6) .LT. V(13,3))GO TO 378
  390 V(7,6)=V(13,3)-V(1,1)
      V(2,1)=V(13,3)
      V(4,1)=D1+V(13,3)-TRTM
      V(13,3)=V(13,3)+V(3,5)
  400 KOUNT = 1
      GO TO 391
C INCREMENT TIMES
  378 DO 379 I=1,3,2
  379 V(I+1,1)=V(I,1)+V(7,6)
391   V(8,5)=V(7,6)*V(7,6)
      IF (V(2,1).GE.V(13,3))  V(13,3)=V(13,3)+V(3,5)
C ESTIMATE VIRTUAL MASS FINAL POSITION AND MAGNITUDE
      V(6,1)=V(5,1)+V(7,1)*V(7,6)+V(8,6)*V(8,5)
      DO 380 J=2,4
  380 V(6,J)=V(5,J)+V(7,J)*V(7,6)+V(10,J+3)*V(8,5)
      RETURN
      END
```

```
      SUBROUTINE  EULMX  (ALP,NN,BET,MM,GAM,LL,P)
      ALP  = FIRST ROTATION ANGLE
      BET  = SECOND ROTATION ANGLE
      GAM  = THIRD ROTATION ANGLE
      NN,MM,LL = AXES OF ROTATION
      IF NN,MM,LL ARE NEGATIVE = MINUS ROTATION
      IF NN = -, MINUS ROTATION ABOUT THAT AXIS
      IF MM = -, MINUS ROTATION ABOUT THAT AXIS
      IF LL = -, MINUS ROTATION ABOUT THAT AXIS
      P = LOCATION OF FIRST ELEMENT OF ROTATION MATRIX
      DIMENSION  A(3,3),P(3,3),F(3,3),G(3,3),H(3,3),D(3,3)
      N = 3
      ALPHA = ALP
      NAXIS = NN
   13 DO 10  I = 1,3
      DO 10  J = 1,3
   10 A(I,J) = 0.
      IF(NAXIS) 6,52,12
    6 ALPHA  =  -ALPHA
      NAXIS  =  -NAXIS
   12 GO TO (20,30,40,53),NAXIS
   53 RETURN
   20 A(1,1) = 1.
      A(2,2) = COS (ALPHA)
      A(2,3) = SIN (ALPHA)
      A(3,2) = -A(2,3)
      A(3,3) =  A(2,2)
      GO TO 21
   30 A(1,1) = COS (ALPHA)
      A(3,1) = SIN (ALPHA)
      A(2,2) =  1.
      A(1,3) = -A(3,1)
      A(3,3) =  A(1,1)
      GO TO 21
   40 A(1,1) = COS (ALPHA)
      A(1,2) = SIN (ALPHA)
      A(2,1) = -A(1,2)
      A(2,2) =  A(1,1)
      A(3,3) =  1.
   21 DO 27  I = 1,3
      DO 27 J = 1,3
      IF(N-2) 26,24,22
   22 F(I,J) = A(I,J)
      GO TO 27
   24 G(I,J) = A(I,J)
      GO TO 27
   26 H(I,J) = A(I,J)
   27 CONTINUE
   45 N = N - 1
      IF (N-1) 50,48,46
   46 IF (MM)  47,52,47
   47 NAXIS = MM
      ALPHA = BET
      GO TO 13
   48 IF (LL) 49,51,49
   49 NAXIS = LL
      ALPHA = GAM
      GO TO 13
      DO 60 I = 1,3
      DO 60 J = 1,3
```

```
      D(I,J) = 0.0
      DO 60 K = 1,3
60    D(I,J) = D(I,J) + H(I,K)*G(K,J)
      DO 65 I = 1,3
      DO 65 J = 1,3
      P(I,J) = 0.0
      DO 65 K = 1,3
65    P(I,J) = P(I,J) + D(I,K)*F(K,J)
      GO TO 54
51    DO 70 I = 1,3
      DO 70 J = 1,3
      P(I,J) = 0.0
      DO 70 K = 1,3
70    P(I,J) = P(I,J) + G(I,K)*F(K,J)
      GO TO 54
   52 DO 55  I = 1,3
      DO 55  J = 1,3
   55 P(I,J) = F(I,J)
   54 RETURN
      END
```

```
      SUBROUTINE GHA
C     THIS SUBROUTINE COMPUTES THE GREENWICH HOUR ANGLE AND THE
C     UNIVERSAL TIME (IN DAYS) WHICH IS USED IN THE TRACKING MODULE
C     TO ORIENT THE STATIONS ON THE EARTH
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      EQMEG = 360.985608288
C  EOMEG IS EARTH ROTATION RATE IN DEG/DAY
      REFJD=2433282.5
C  REFERENCE JULIAN DATE IS 0-HRS,JAN-1, 1950
      TSTAR=DATEJ+2415020.-REFJD
      ID = TSTAR
      D=ID
      TFRAC=TSTAR-D
      GH=100.0755426+D*(0.985647346+D*2.9015E-13) +EQMEG*TFRAC
1     IF(GH)2,3,3
2     GH=GH+360.
      GO TO 1
3     IF(GH-360.) 5,4,4
4     GH=GH-360.
      GO TO 3
5     CONTINUE
      UNIVT= (GH)/EQMEG
      RETURN
      END
```

```
      SUBROUTINE GUID(RF,IGP,TEVN,GA,ADA)
C
C
C     THIS SUBROUTINE COMPUTES THE GAMMA MATRIX FOR USE IN THE GUIDANCE
C     MODULE
C
C     INPUT ARGUMENTS
C        RF   -- POSITION AND VELOCITY OF SPACECRAFT AT TIME TEVN
C        IGP  -- GUIDANCE POLICY CODE
C                =1 -- FIXED TIME OF ARRIVAL
C                =2 -- TWO VARIABLE B-PLANE
C                =3 -- THREE VARIABLE B-PLANE
C        TEVN -- TIME OF GUIDANCE EVENT
C
C     OUTPUT ARGUMENT
C        GA   -- GAMMA MATRIX
C
C     THIS SUBROUTINE REQUIRES THE USE OF THE FOLLOWING SUBROUTINES
C        NTM
C        PSIM
C        PARTL
C        VARADA
C        BLOCK DATA
C
C
      DIMENSION RI(6),RF(6),GA(3,6),XCA(6),XSIP(3),XSIV(3),RTPS(6),
     $PHI1(3,3),PHI2(3,3),PBT(6),PBR(6),A(2,3),BB(2,3),ADA(3,6)
      DIMENSION PHI3(2,2),EGVL(3),EGVCT(3,3),DUM1(2,2)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      MAX=60
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      GO TO (20,210,210), IGP
 10   IF (IPOL) 50,20,50
 20   DELTM=FNTM-TEVN
      IF(ICL.NE.0) GO TO 600
      DO 30 I=1,6
 30   RI(I)=RF(I)
      TRTM1=TEVN
      ICS=ICL2
```

```
      ICL2=NTP
      IPR=IPRINT
      IPRINT=1
      CALL NTM(RI,RF,NTMC,-1)
      TSOI1=DSI-DATEJ
      ICL2=ICS
      IPRINT=IPR
      IF(ISPH.EQ.1) GO TO 35
      WRITE(6,1000)
      GO TO 500
35    TCA=DC-DATEJ
      D=TCA+DATEJ
      NO(1)=NTP
      CALL ORB(NTP,D)
      CALL EPHEM(1,D,1)
      DO 40 I=1,3
      XCA(I)=RC(I)+XP(I)*ALNGTH
40    XCA(I+3)=RC(I+3)+XP(I+3)*ALNGTH/TM
      GO TO 70
50    DO 60 I=1,6
60    RI(I)=RF(I)
70    DELTM=TCA-TEVN
      TRTM1=TEVN
      WRITE(6,2000) TCA,RC
      LINES=21
      NO(1)=NTP
      CALL ORB(NTP,DSI)
      CALL EPHEM(1,DSI,1)
      DO 100 I=1,3
      RF(I)=RSI(I)+XP(I)*ALNGTH
100   RF(I+3)=VSI(I)+XP(I+3)*ALNGTH/TM
      DELTM=DSI-DATEJ-TEVN
      CALL PSIM(RI,RF,ISTMC)
      DO 102 I=1,NDIM
      DO 102 J=1,NDIM
102   PSIP(I,J)=PSI(I,J)
      DELTM=TCA-(DSI-DATEJ)
      D1=DTMAX
      DTMAX=300.
      CALL PSIM(RF,XCA,ISTMC)
      DTMAX=D1
      DO 103 I=1,NDIM
      DO 103 J=1,NDIM
103   Q(I,J)=PSI(I,J)
      DO 104 I=1,NDIM
      DO 104 J=1,NDIM
      PSI(I,J)=0.
      DO 104 K=1,NDIM
104   PSI(I,J)=PSI(I,J)+Q(I,K)*PSIP(K,J)
      IF (LINES.LT.MAX-8 ) GO TO 71
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
71    WRITE (6,3002) TCA,TEVN
      LINES=LINES+5
      DO 72 I=1,NDIM
      IF (LINES.LT.MAX-4 ) GO TO 73
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
```

```
73    IF (NDIM.EQ.6) GO TO 74
      WRITE (6,3003)I
      LINES=LINES+1
74    WRITE(6,3011) (PSI(I,J),J=1,NDIM)
72    LINES=LINES+(NDIM-1)/6+1
      DO 90 I=1,3
      DO 90 J=1,3
      ADA (I,J)=PSI(I,J)
90    ADA(I,J+3)=PSI(I,J+3)
      CALL PARTL(RSI,VSI,B1,BDT1,BDR1,PBT,PBR)
      DO 91 J=1,6
      EM(1,J)=PBT(J)
91    EM(2,J)=PBR(J)
      GO TO 407
200   IF (IIPOL) 240,210,240
210   DELTM=FNTM-TEVN
      DO 220 I=1,6
220   RI(I)=RF(I)
      TRTM1=TEVN
      IF(ISPH.NE.0) GO TO 225
      IPR=IPRINT
      IPRINT=1
      ISP=ISP2
      ISP2=NTP
      CALL NTM(RI,RF,NTMC,-1)
      TSOI1=DSI-DATEJ
      ISP2=ISP
      IPRINT=IPR
      TSI=DSI-DATEJ
      IF(ISPH.EQ.1) GO TO 221
      WRITE(6,1000)
1000  FORMAT(///8X*VEHICLE DID NOT REACH SPHERE OF INFLUENCE BEFORE FINA
     $L TRAJECTORY TIME.*/8X*RETURNING TO BASIC CYCLE*///)
      GO TO 500
221   D=DSI
      NO(1)=NTP
      CALL ORB(NTP,D)
      CALL EPHEM(1,D,1)
      DO 230 I=1,3
      XSIP(I)=RSI(I)
      XSIV(I)=VSI(I)
      RTPS(I)=RSI(I)+XP(I)*ALNGTH
230   RTPS(I+3)=VSI(I)+XP(I+3)*ALNGTH/TM
      BS=B
      BDTS=BDT
      BDRS=BDR
      GO TO 260
240   DO 250 I=1,6
250   RI(I)=RF(I)
260   IF(IGP.EQ.3) GO TO 400
      CALL PARTL(XSIP,XSIV,B1,BDT1,BDR1,PBT,PBR)
      DO 261 J=1,6
      EM(1,J)=PBT(J)
261   EM(2,J)=PBR(J)
      TRTM1=TEVN
      ISPH=1
      DELTM=TSI-TEVN
      CALL PSIM(RI,RTPS,ISTMC)
      WRITE(6,3005) TSI,XSIP,XSIV,BS,BDTS,BDRS
      LINES=23
```

```
      IF (LINES.LT.MAX-8 ) GO TO 266
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
266   WRITE (6,3002) TSI,TEVN
      DO 267 I=1,NDIM
      IF (LINES.LT.MAX-4 ) GO TO 268
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
268   IF (NDIM.EQ.6) GO TO 269
      WRITE(6,3003) I
      LINES=LINES+1
269   WRITE(6,3011) (PSI(I,J),J=1,NDIM)
267   LINES=LINES+(NDIM-1)/6+1
      IF (LINES.LT.MAX-7 ) GO TO 270
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
270   WRITE (6,3010) PBT,PBR
      LINES=LINES+7
      DO 280 I=1,3
      A(1,I)=0.
      BB(1,I)=0.
      DO 280 J=1,6
      A(1,I)=A(1,I)+PBT(J)*PSI(J,I)
280   BB(1,I)=BB(1,I)+PBT(J)*PSI(J,I+3)
      DO 290 I=1,3
      A(2,I)=0.
      BB(2,I)=0.
      DO 290 J=1,6
      A(2,I)=A(2,I)+PBR(J)*PSI(J,I)
290   BB(2,I)=BB(2,I)+PBR(J)*PSI(J,I+3)
      IF (LINES.LT.MAX-12) GO TO 271
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
271   WRITE (6,3006) ((A(I,J),J=1,3),I=1,2),((BB(I,J),J=1,3),I=1,2)
      LINES=LINES+12
      DO 272 I=1,2
      DO 272 J=1,3
      ADA(I,J)=A(I,J)
272   ADA(I,J+3)=BB(I,J)
      IF(LINES.LT.MAX-6) GO TO 273
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
273   DO 274 I=1,2
      DO 274 J=1,2
      PHI3(I,J)=0.
      DO 274 K=1,6
      DO 274 L=1,6
274   PHI3(I,J)=PHI3(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
      WRITE(6,3016) ((PHI3(I,J),J=1,2),I=1,2)
      LINES=LINES+7
      K=0
      DO 275 J=1,2
      DO 275 I=1,2
      K=K+1
275   PBT(K)=PHI3(I,J)
```

```
      IF(LINES.LT.MAX-16) GO TO 276
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
276   CALL JACOBI(PBT,EGVL,DUM1, 2,FOV)
      WRITE(6,3013) (I,EGVL(I),I=1,2)
      WRITE(6,3017) (I,( DUM1(I,J),J=1,2),I=1,2)
      IF(IHYP1.EQ.2) GO TO 278
      IF(LINES.LT.MAX-9) GO TO 277
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
277   CALL HYELS(1,PHI3,2)
      LINES=LINES+8
278   IF(IHYP1.EQ.1) GO TO 281
      IF(LINES.LT.MAX-9) GO TO 279
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
279   CALL HYELS(3,PHI3,2)
      LINES=9
      LINES=LINES+8
281   DO 310 I=1,2
      DO 310 J=1,2
      PHI3(I,J)=0.
      DO 310 K=1,3
310   PHI3(I,J)=PHI3(I,J)+BB(I,K)*BB(J,K)
      CALL MATIN(PHI3,PHI3,2)
      DO 320 I=1,3
      DO 320 J=1,2
      PHI2(I,J)=0.
      DO 320 K=1,2
320   PHI2(I,J)=PHI2(I,J)+BB(K,I)*PHI3(K,J)
325   DO 330 I=1,3
      DO 330 J=1,3
      GA(I,J)=0.
      DO 330 K=1,2
330   GA(I,J)=GA(I,J)-PHI2(I,K)*A(K,J)
      DO 340 I=1,3
      DO 340 J=4,6
      GA(I,J)=0.
      DO 340 K=1,2
340   GA(I,J)=GA(I,J)-PHI2(I,K)*BB(K,J-3)
      IF (LINES.LT.MAX-7 ) GO TO 341
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
341   WRITE(6,3007) ((GA(I,J),J=1,6),I=1,3)
      IIPOL=1
      GO TO 500
400   CALL VARADA(RI,XSIP,XSIV,TEVN,TSI,ADA,BS,BDTS,BDRS)
      IF(ISPH.EQ.0) GO TO 500
      WRITE(6,3005) TSI,XSIP,XSIV,BS,BDTS,BDRS
      CALL PARTL(XSIP,XSIV,B1,BDT1,BDR1,PBT,PBR)
      DO 408 J=1,6
      EM(1,J)=PBT(J)
408   EM(2,J)=PBR(J)
407   WRITE (6,3008) ((ADA(I,J),J=1,6),I=1,3)
      DO 401 I=1,3
      DO 401 J=1,3
      PHI1(I,J)=0.
```

```
      DO 401 K=1,6
      DO 401 L=1,6
401   PHI1(I,J)=PHI1(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
      WRITE(6,3012) ((PHI1(I,J),J=1,3),I=1,3)
      DO 402 I=1,3
      DO 402 J=1,3
402   PHI2(I,J)=PHI1(I,J)
      CALL JACOBI(PHI2,EGVL,EGVCT,3,FOV)
      WRITE(6,3013) (I,EGVL(I),I=1,3)
      WRITE(6,3014) (I,(EGVCT(I,J),J=1,3),I=1,3)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      IF(IHYP1-2) 403,404,403
403   CALL HYELS(1,PHI1,3)
404   IF(IHYP1-1) 405,406,405
405   CALL HYELS(3,PHI1,3)
406   DO 410 I=1,3
      DO 410 J=1,3
      PHI1(I,J)=ADA(I,J)
410   PHI2(I,J)=ADA(I,J+3)
      CALL MATIN(PHI2,PHI2,3)
      DO 420 I=1,3
      DO 420 J=1,3
      GA(I,J)=0.
      DO 420 K=1,3
420   GA(I,J)=GA(I,J)-PHI2(I,K)*PHI1(K,J)
      DO 430 I=1,3
      DO 430 J=4,6
430   GA(I,J)=0.
      GA(1,4)=-1.
      GA(2,5)=-1.
      GA(3,6)=-1.
      IF(IGP.EQ.1) GO TO 111
      WRITE (6,3009) ((GA(I,J),J=1,6),I=1,3)
440   IIPOL=1
      GO TO 500
111   WRITE(6,3004) ((GA(I,J),J=1,6),I=1,3)
      IPOL=1
      GO TO 500
600   WRITE(6,1001)
1001  FORMAT(///8X*CLOSEST APPROACH HAS BEEN PREVIOUSLY ENCOUNTERED*/
     $8X*RETURNING TO BASIC CYCLE*)
      ISPH=0
      GO TO 500
225   WRITE(6,3015)
      ISPH=0
500   RETURN
3000  FORMAT (1H1//8X2A10*--  GUIDANCE EVENT  AT TRAJECTORY TIME *F12.3,
     $* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*)//)
2000  FORMAT(      8X*TIME OF CLOSEST APPROACH *F12.3//8X*AT CLOSEST APP
     $ROACH*/8X*POSITION*3E20.10/8X*VELOCITY*3E20.10)
3002  FORMAT(///8X*STATE TRANSITION MATRIX RELATING STATE VECTOR AT TIME
     $ *F12.3* DAYS TO THAT AT TIME *F12.3* DAYS*/)
3003  FORMAT(10X,*ROW*I3)
3004  FORMAT(///8X*GUIDANCE MATRIX---FIXED TIME OF ARRIVAL GUIDANCE POLI
     $CY*/3(8X6E20.10/))
3005  FORMAT (      8X*TIME AT WHICH VEHICLE REACHES SPHERE OF INFLUENCE
     $ OF TARGET PLANET *F12.3* DAYS*//8X*AT SPHERE OF INFLUENCE*//8X*PO
     $SITION*3E20.10/8X*VELOCITY*3E20.10//8X*B = *E20.10,5X*B DOT T = *E
     $20.10,5X*B DOT R = *E20.10)
```

```
3006  FORMAT(///8X*A*/2(8X3E20.10/)8X*B*/2(8X3E20.10/))
3007  FORMAT(///8X*GUIDANCE MATRIX---TWO VARIABLE B-PLANE GUIDANCE POLIC
     *Y*/3(8X6E20.10/))
3008  FORMAT(///8X*VARIATION MATRIX*/3(8X6E20.10/))
3009  FORMAT(///8X*GUIDANCE MATRIX---THREE VARIABLE B-PLANE GUIDANCE POL
     *ICY*/3(8X6E20.10/))
3010  FORMAT(///8X*PBT*/8X6E20.10/8X*PBR*/8X6E20.8)
3011  FORMAT(8X6E20.10)
3012  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITIONS BEFORE CORRECTION*/
     $3(8X3E20.10/))
3013  FORMAT(///20X*EIGENVALUES OF ABOVE MATRIX*/3(22XI2E20.10/))
3014  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.10/))
3015  FORMAT(///8X*SPHERE OF INFLUENCE HAS BEEN PREVIOUSLY ENCOUNTERED*
     $/RETURNING TO BASIC CYCLE*)
3016  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITIONS BEFORE CORRECTION*/
     $2(8X2E20.10/))
3017  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/2(22XI2,2E20.10/))
      END
```

```
      SUBROUTINE GUIDM(RI,TEVN)
C
C
C     THIS SUBROUTINE CONTAINS THE LOGIC FOR THE GUIDANCE EVENT.
C
C     GUIDM USES THE FOLLOWING SUBROUTINES
C        NTM
C        PSIM
C        DYNO
C        NAVM
C        GUID
C
C
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),GA(3,6),S(3,3),P1(17,17),RF(6),P2(6,6)
      DIMENSION GAP(3,6),EGVL(3),EGVCT(3,3),EXV(3),EXEC(3,3)
      DIMENSION VEIG(9),ADA(3,6),DUM(2,2)
C     CALCULATE P(TEVN,TRTM1)
      MAX=60
      DELTM=TEVN-TRTM1
      ITEMP=(NDIM-1)/6+1
      CALL NTM(RI,RF,NTMC,1)
401   ISPHC=ISPH
      DO 5 I=1,6
5     XF(I)=RF(I)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
      WRITE(6,3012) TEVN
      WRITE(6,3001) (CMPNM(IAUG,I),XF(I),I=1,NDIM)
      LINES=LINES+NDIM+1
      CALL PSIM(RI,RF,ISTMC)
      WRITE(6,3003) TEVN,TRTM1
      LINES=LINES+5
      DO 2 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 1
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
1     IF (NDIM.EQ.6)GO TO 13
      WRITE (6,3004)I
```

```
      LINES=LINES+1
13    WRITE (6,3005) (PSI(I,J),J=1,NDIM)
2     LINES=LINES+ITEMP
      CALL DYNO(0)
      IF(LINES.LT.MAX-8) GO TO 3
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
3     WRITE (6,3006)
      WRITE (6,3005) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF(LINES.LT.MAX-8) GO TO 4
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
4     WRITE (6,3007) TEVN,TRTM1
      LINES=LINES+5
      DO 6 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 14
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
14    IF (NDIM.EQ.6) GO TO 15
      WRITE (6,3004) I
      LINES=LINES+1
15    WRITE (6,3005) (P(I,J),J=1,NDIM)
6     LINES=LINES+ITEMP
      ICODE2=1
199   ICODE=0
      K=0
      DO 200 J=1,3
      DO 200 I=1,3
      S(I,J)=P(I,J)
      K=K+1
200   VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF(LINES.LT.MAX-16) GO TO 201
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
201   WRITE(6,2000) (I,EGVL(I),I=1,3)
      WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
210   IF(IHYP1-2) 220,230,220
220   IF(LINES.LT.MAX-16) GO TO 221
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
221   CALL HYELS(1,S,3)
      LINES=LINES+16
230   IF(IHYP1-1) 240,250,240
240   IF(LINES.LT.MAX-16) GO TO 241
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
241   CALL HYELS(3,S,3)
      LINES=LINES+16
250   IF(ICODE)260,260,290
260   IF(IEIG) 290,290,270
```

```
270     K=0
        DO 280 J=1,3
        DO 280 I=1,3
        K=K+1
        S(I,J)=P(I+3,J+3)
280     VEIG(K)=S(I,J)
        CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
        IF(LINES.LT.MAX-16) GO TO 281
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
        LINES=9
281     WRITE(6,2002) (I,EGVL(I),I=1,3)
        WRITE(6,2003) (I,(EGVCT(I,J),J=1,3),I=1,3)
        LINES=LINES+16
        ICODE=1
        GO TO 210
290     GO TO (300,310,320), ICODE2
300     DO 10 I=1,NDIM
        DO 10 J=1,NDIM
        P1(I,J)=P(I,J)
10      P(I,J)=PG(I,J)
        DO 20 I=1,6
20      RI(I)=XG(I)
C       CALCULATE P(TEVN,TKC-1)
        DELTM=TEVN-TG
        TRTM1=TG
        CALL PSIM(RI,RF,ISTMC)
        IF(LINES.LT.MAX-8) GO TO 7
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
        LINES=9
7       WRITE (6,3003) TEVN,TG
        LINES=LINES+5
        DO 8 I=1,NDIM
        IF(LINES.LT.MAX-4) GO TO 16
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
        LINES=9
16      IF (NDIM.EQ.6) GO TO 17
        WRITE (6,3004) I
        LINES=LINES+1
17      WRITE (6,3005) (PSI(I,J),J=1,NDIM)
8       LINES=LINES+ITEMP
        CALL DYNO(0)
        IF(LINES.LT.MAX-8) GO TO 9
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
        LINES=9
9       WRITE(6,3006)
        WRITE(6,3005) (Q(I,I),I=1,NDIM)
        LINES=LINES+8
        CALL NAVM(1,1)
        IF(LINES.LT.MAX-9) GO TO 11
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
        LINES=9
11      WRITE (6,3008) TEVN,TG
        LINES=LINES+5
        DO 12 I=1,NDIM
        IF(LINES.LT.MAX-4) GO TO 18
        IPGN=IPGN+1
        WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
```

```
      LINES=9
18    IF(NDIM.EQ.6) GO TO 19
      WRITE(6,3004) I
      LINES=LINES+1
19    WRITE(6,3005) (P(I,J),J=1,NDIM)
12    LINES=LINES+ITEMP
      ICODE2=2
      GO TO 199
310   DO 30 I=1,6
      DO 30 J=1,6
30    P2(I,J)=P(I,J)
      NGE=NGE+1
      IGP=ICDT3(NGE)
      IQP=ICDQ3(NGE)
      CALL GUID(RF,IGP,TEVN,GA,ADA)
      IF(ISPH.EQ.0) GO TO 105
31    DO 50 I=1,3
      DO 50 J=1,6
      GAP(I,J)=0.
      DO 50 K=1,6
50    GAP(I,J)=GAP(I,J)+GA(I,K)*P2(K,J)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
      DO 60 I=1,3
      DO 60 J=1,3
      S(I,J)=0.
      DO 60 K=1,6
60    S(I,J)=S(I,J)+GAP(I,K)*GA(J,K)
      WRITE(6,3009) ((S(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      IF(IQP.NE.0) GO TO 40
      TRS=S(1,1)+S(2,2)+S(3,3)
      U=S(1,1)*(S(2,2)+S(3,3))+S(2,2)*S(3,3)-S(1,2)*S(1,2)-S(1,3)*S(1,3)
     $-S(2,3)*S(2,3)
      RHO=SQRT(2.*TRS/3.1415926)*(1.+U*(1.1415926)/(TRS*TRS*2.3238))
      SDV=SQRT(TRS-RHO*RHO)
      WRITE(6,3013) RHO
      WRITE(6,3017) SDV
      LINES=LINES+2
      DUM1=S(1,1)+S(2,2)
      EXEC(1,1)=S(1,1)*(SIGPRO+SIGRES/TRS)+S(2,2)*(TRS*SIGALP/DUM1)
     $+S(1,1)*S(3,3)*SIGBET/DUM1
      EXEC(1,2)=S(1,2)*(SIGPRO+SIGRES/TRS-TRS*SIGALP/DUM1+S(3,3)*SIGBET/
     $DUM1)
      EXEC(2,1)=EXEC(1,2)
      EXEC(1,3)=S(1,3)*(SIGPRO+SIGRES/TRS-SIGBET)
      EXEC(3,1)=EXEC(1,3)
      EXEC(2,2)=S(2,2)*(SIGPRO+SIGRES/TRS)+S(1,1)*TRS*SIGALP/DUM1+S(2,2)
     $*S(3,3)*SIGBET/DUM1
      EXEC(2,3)=S(2,3)*(SIGPRO+SIGRES/TRS-SIGBET)
      EXEC(3,2)=EXEC(2,3)
      EXEC(3,3)=S(3,3)*(SIGPRO+SIGRES/TRS)+DUM1*SIGBET
      GO TO 75
40    TRS=S(1,1)+S(2,2)+S(3,3)
      U=S(1,1)*(S(2,2)+S(3,3))+S(2,2)*S(3,3)-S(1,2)*S(1,2)-S(1,3)*S(1,3)
     $-S(2,3)*S(2,3)
      RHO=SQRT(2.*TRS/3.1415926)*(1.+U*(1.1415926)/(TRS*TRS*2.3238))
      SDV=SQRT(TRS-RHO*RHO)
      WRITE(6,3013) RHO
```

```
      WRITE(6,3017) SDV
      LINES=LINES+2
      CALL JACOBI(S,EGVL,EGVCT,3,FOV)
      WRITE(6,1000) (I,EGVL(I),I=1,3)
      WRITE (6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      AMAX=EGVL(1)
      MAP=1
      IF(AMAX.GE.EGVL(2)) GO TO 70
      AMAX=EGVL(2)
      MAP=2
70    IF(AMAX.GE.EGVL(3)) GO TO 71
      AMAX=EGVL(3)
      MAP=3
71    EGM=SQRT(EGVCT(MAP,1)*EGVCT(MAP,1)+EGVCT(MAP,2)*EGVCT(MAP,2)+
     $EGVCT(MAP,3)*EGVCT(MAP,3))
      DUM = RHO/EGM
      EXV(1)=EGVCT(MAP,1)*DUM
      EXV(2)=EGVCT(MAP,2)*DUM
      EXV(3)=EGVCT(MAP,3)*DUM
      IF(LINES.LT.MAX-5) GO TO 72
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
72    WRITE(6,3015) EXV
      LINES=LINES+5
      X2=EXV(1)*EXV(1)
      Y2=EXV(2)*EXV(2)
      Z2=EXV(3)*EXV(3)
      DUM1=X2+Y2
      EXM=DUM1+Z2
      DUM2=SIGRES/EXM
      EXEC(1,1)=X2*(SIGPRO+DUM2)+Y2*EXM*SIGALP/DUM1+X2*Z2*SIGBET/DUM1
      EXEC(1,2)=EXV(1)*EXV(2)*(SIGPRO+DUM2-EXM*SIGALP/DUM1+Z2*SIGBET/
     $DUM1)
      EXEC(2,1)=EXEC(1,2)
      EXEC(1,3)=EXV(1)*EXV(3)*(SIGPRO+DUM2-SIGBET)
      EXEC(3,1)=EXEC(1,3)
      EXEC(2,2)=Y2*(SIGPRO+DUM2)+X2*EXM*SIGALP/DUM1+Y2*Z2*SIGBET/DUM1
      EXEC(2,3)=EXV(2)*EXV(3)*(SIGPRO+DUM2-SIGBET)
      EXEC(3,2)=EXEC(2,3)
      EXEC(3,3)=Z2*(SIGPRO+DUM2)+DUM1*SIGBET
75    DO 80 I=1,NDIM
      DO 80 J=1,NDIM
      P(I,J)=P1(I,J)
80    PG(I,J)=P1(I,J)
      WRITE(6,3010) ((EXEC(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      DO 81 I=1,3
      DO 81 J=1,3
81    S(I,J)=EXEC(I,J)
      ICODE2=1
88    IF(LINES.LT.MAX-16) GO TO 82
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
82    CALL JACOBI(S,EGVL,EGVCT,3,FOV)
      WRITE(6,1000) (I,EGVL(I),I=1,3)
      WRITE(6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
```

```
      IF(IHYP1.EQ.2) GO TO 84
      IF(LINES.LT.MAX-16) GO TO 83
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
83    CALL HYELS(1,EXEC,3)
      LINES=LINES+16
84    IF(IHYP1.EQ.1) GO TO 86
      IF(LINES.LT.MAX-16) GO TO 85
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
85    CALL HYELS(3,EXEC,3)
      LINES=LINES+16
86    GO TO (87,330) , ICODE2
87    DO 90 I=1,3
      DO 90 J=1,3
      DUM = P(I+3,J+3) + EXEC(I,J)
      P(I+3,J+3)=DUM
90    PG(I+3,J+3)=DUM
      IF(LINES.LT.MAX-8) GO TO 91
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
91    WRITE(6,3011) TEVN,TEVN
      LINES=LINES+5
      DO 92 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 93
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
93    IF (NDIM.EQ.6) GO TO 94
      WRITE(6,3004) I
      LINES=LINES+1
94    WRITE(6,3005) (P(I,J),J=1,NDIM)
92    LINES=LINES+ITEMP
      ICODE2=3
      GO TO 199
320   NN=3
      IF(IGP.EQ.2) NN=2
      DO 321 I=1,NN
      DO 321 J=1,NN
      S(I,J)=0.
      DO 321 K=1,6
      DO 321 L=1,6
      S(I,J)=S(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
321   EXEC(I,J)=S(I,J)
      IF(LINES.LT.MAX-8) GO TO 328
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
328   IF(IGP.EQ.2) GO TO 322
      WRITE(6,3014) ((S(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      ICODE2=2
      GO TO 88
322   WRITE(6,3016) ((S(I,J),J=1,2),I=1,2)
      LINES=LINES+8
      IF(LINES.LT.MAX-16) GO TO 323
      IPGN=IPGN+1
```

```
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
323   K=0
      DO 324 J=1,2
      DO 324 I=1,2
      K=K+1
324   VEIG(K)=S(I,J)
      CALL JACOBI(VEIG,EGVL,DUM  ,2,FOV)
      WRITE(6,1000) (I,EGVL(I),I=1,2)
      WRITE(6,1002) (I,(  DUM(I,J),J=1,2),I=1,2)
      LINES=LINES+14
      IF(IHYP1.EQ.2) GO TO 326
      IF(LINES.LT.MAX-9 ) GO TO 325
      WRITE(6,3000)  (MDNM(ITR,K),K=1,2),TEVN,IPROB
      LINES=9
325   CALL HYELS(1,EXEC,2)
      LINES=LINES+8
326   IF(IHYP1.EQ.1) GO TO 330
      IF(LINES.LT.MAX-9 ) GO TO 327
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,I),I=1,2),TEVN,IPROB,IPGN
      LINES=9
327   CALL HYELS(3,EXEC,2)
      LINES=LINES+8
330   DO 100 I=1,6
      XG(I)=XF(I)
100   XI(I)=XF(I)
      DO 101 I=7,NDIM
101   XI(I)=XF(I)
      TG=TEVN
      TRTM1=TEVN
      ISPH=ISPHC
      RETURN
105   DO 120 I=1,NDIM
120   XI(I)=XF(I)
      TRTM1=TEVN
      DO 125 I=1,NDIM
      DO 125 J=1,NDIM
125   P(I,J)=P1(I,J)
      ISPH=ISPHC
      RETURN
1000  FORMAT(///20X*EIGENVALUES OF ABOVE MATRIX*/3(22X,I2,E20.10/))
1001  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/3(22X,I2,3E20.10/))
1002  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/2(22X,I2,2E20.10/))
2000  FORMAT(///20X*POSITION EIGENVALUES OF ABOVE MATRIX*/3(22XI2E20.10/
     $))
2001  FORMAT(///20X*POSITION EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.
     $10/))
2002  FORMAT(///20X*VELOCITY EIGENVALUES OF ABOVE MATRIX*/3(22XI2E20.10/
     $))
2003  FORMAT(///20X*VELOCITY EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.
     $10/))
3000  FORMAT (1H1//8X2A10*--  GUIDANCE EVENT  AT TRAJECTORY TIME *F12.3,
     $* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*)//)
3001  FORMAT(10X,A10,E20.13)
3002  FORMAT (1H1)
3003  FORMAT (///8X*STATE TRANSITION MATRIX  -- PSI(*F12.3*,*F12.3*)*/)
3004  FORMAT(10X*ROW*I3)
3005  FORMAT(16X,6E17.8)
3006  FORMAT (///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
```

```
3007  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT  --  P(*
     $F12.3*,*F12.3*)*/)
3008  FORMAT (///8X*COVARIANCE MATRIX RELATING THE TIME OF THIS GUIDANCE
     $ EVENT TO THAT OF THE LAST GUIDANCE EVENT -- P(*F12.3*,*F12.3*)*/)
3009  FORMAT(///8X*COVARIANCE MATRIX ASSOCIATED WITH VELOCITY COMPONENTS
     **/3(8X3E20.10/))
3010  FORMAT(///8X*EXECUTION ERROR MATRIX*/3(12X,3E20.10/))
3011  FORMAT(///8X*MODIFIED COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT
     $ -- P(*F12.3*,*F12.3*)*/)
3012  FORMAT(8X*STATE VECTOR AT TIME * F12.3* DAYS*/)
3013  FORMAT(/10X*EXPECTED VALUE OF DELTA V. . .*E20.10)
3014  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION*/
     $3(10X,3E20.10/))
3015  FORMAT(///8X*EXPECTED VALUE OF VELOCITY CORRECTION*/8X3E20.10)
3016  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION*/
     $2(10X,2E20.10/))
3017  FORMAT( /10X*STANDARD DEVIATION OF EXPECTED VALUE OF DELTA V. . .*
     $E20.10)
      END
```

```
      SUBROUTINE GUIS(RF,RF1,IGP,TEVN,GA,ADA)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON /CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6), RI1(6),RF1(6),GA(3,6),ADA(3,6),XCA(6)
      DIMENSION PHI1(3,3),PHI2(3,3),PBT(6),PBR(6),A(2,3),BB(2,3)
      DIMENSION PHI3(2,2),EGVL(3),EGVCT(3,3),RTPS(6),DUM(2,2),DUM1(2,2)
      MAX=60
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
      GO TO (10,200,200), IGP
10    TRTM1=TEVN
      DELTM=FNTM-TEVN
      IPR=IPRINT
      IPRINT=1
      ICS=ICL2
      ICL2=NTP
      DO 20 I=1,6
20    RI(I)=RF(I)
      CALL NTM(RI,RF,NTMC,-1)
      TSOI1=DSI-DATEJ
      IPRINT=IPR
      ICL2=ICS
      IF (ISPH.EQ.1) GO TO 25
      WRITE(6,3003)
      GO TO 500
25    CALL PARTL(RSI,VSI,B1,BDT1,BDR1,PBT,PBR)
      DO 30 I=1,6
      EM(1,I)=PBT(I)
30    EM(2,I)=PBR(I)
      TCA=DC-DATEJ
      RMCA=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      VMCA=SQRT(RC(4)*RC(4)+RC(5)*RC(5)+RC(6)*RC(6))
      WRITE(6,3040) TCA,(RC(I),I=1,3),RMCA,(RC(I),I=4,6),VMCA
      LINES=LINES+5
      WRITE(6,3041) ((EM(I,J),J=1,6),I=1,2)
      LINES=LINES+8
```

```
      DO 40 I=1,6
40    RI1(I)=RF1(I)
      IF(NQE.NE.0) GO TO 42
      DO 41 I=1,6
41    RF1(I)=RF(I)
      GO TO 43
42    IPR=IPRINT
      ICLS=ICL2
      IPRINT=1
      ICL2=NTP
      CALL NTM(RI1,RF1,NTMC,-2)
      ICL2=ICS
      IPRINT=IPR
43    IF (ISPH.EQ.1) GO TO 50
      WRITE(6,3002)
      GO TO 500
50    NO(1)=NTP
      CALL ORB(NTP,DC)
      CALL EPHEM(1,DC,1)
      DO 60 I=1,3
      XCA(I)=RC(I)+XP(I)*ALNGTH
60    XCA(I+3)=RC(I+3)+XP(I+3)*ALNGTH/TM
      TCA=DC-DATEJ
      RMCA=SQRT(RC(1)*RC(1)+RC(2)*RC(2)+RC(3)*RC(3))
      VMCA=SQRT(RC(4)*RC(4)+RC(5)*RC(5)+RC(6)*RC(6))
      WRITE(6,3001) TCA,(RC(I),I=1,3),RMCA,(RC(I),I=4,6),VMCA
      LINES=LINES+11
      CALL ORB(NTP,DSI)
      CALL EPHEM(1,DSI,1)
      DO 70 I=1,3
      RF1(I)=RSI(I)+XP(I)*ALNGTH
70    RF1(I+3)=VSI(I)+XP(I+3)*ALNGTH/TM
      DELTM=DSI-DATEJ-TEVN
      CALL PSIM(RI1,RF1,ISTMC)
      DO 71 I=1,NDIM
      DO 71 J=1,NDIM
71    PSIP(I,J)=PSI(I,J)
      DELTM=TCA-(DSI-DATEJ)
      D1=DTMAX
      DTMAX=300.
      CALL PSIM(RF1,XCA,ISTMC)
      DTMAX=D1
      DO 72 I=1,NDIM
      DO 72 J=1,NDIM
72    Q(I,J)=PSI(I,J)
      DO 73 I=1,NDIM
      DO 73 J=1,NDIM
      PSI(I,J)=0.
      DO 73 K=1,NDIM
73    PSI(I,J)=PSI(I,J)+Q(I,K)*PSIP(K,J)
      IF(LINES.LT.MAX-9)GO TO 80
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
80    WRITE(6,3004) TCA,TEVN
      LINES=LINES+5
      DO 83 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 81
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
```

```
      LINES=9
81    IF(NDIM.EQ.6) GO TO 82
      WRITE(6,3005) I
      LINES=LINES+1
82    WRITE(6,3006) (PSI(I,J),J=1,NDIM)
83    LINES=LINES+(NDIM-1)/6+1
      DO 90 I=1,3
      DO 90 J=1,6
90    ADA(I,J)=PSI(I,J)
91    IF(LINES.LT.MAX-8 ) GO TO 100
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
100   WRITE(6,3007)((ADA(I,J),J=1,6),I=1,3)
      LINES=LINES+8
      DO 110 I=1,3
      DO 110 J=1,3
      PHI1(I,J)=0.
      DO 110 K=1,6
      DO 110 L=1,6
110   PHI1(I,J)=PHI1(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
      IF (LINES.LT.MAX-8) GO TO 120
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
120   WRITE(6,3008) ((PHI1(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      DO 130 I=1,3
      DO 130 J=1,3
130   PHI2(I,J)=PHI1(I,J)
      CALL JACOBI(PHI2,EGVL,EGVCT,3,FOV)
      IF(LINES.LT.MAX-14) GO TO 131
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
131   WRITE(6,3009) (I,EGVL(I),I=1,3)
      WRITE(6,3010) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+14
      IF(IHYP1-2) 140,150,140
140   IF(LINES.LT.MAX-16) GO TO 141
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
141   CALL HYELS(1,PHI1,3)
      LINES=LINES+16
150   IF (IHYP1-1) 151,160,151
151   IF (LINES.LT.MAX-16) GO TO 152
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
152   CALL HYELS(3,PHI1,3)
      LINES=LINES+16
160   DO 161 I=1,3
      DO 161 J=1,3
      PHI1(I,J)=ADA(I,J)
161   PHI2(I,J)=ADA(I,J+3)
      CALL MATIN(PHI2,PHI2,3)
      DO 170 I=1,3
      DO 170 J=1,3
      GA(I,J)=0.
```

```
      DO 170 K=1,3
170   GA(I,J)=GA(I,J)-PHI2(I,K)*PHI1(K,J)
      DO 180 I=1,3
      DO 180 J=4,6
180   GA(I,J)=0.
      GA(1,4)=-1.
      GA(2,5)=-1.
      GA(3,6)=-1.
      IF(LINES.LT.MAX-8) GO TO 190
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
190   IF(IGP.EQ.3) GO TO 191
      WRITE(6,3011) ((GA(I,J),J=1,6),I=1,3)
      LINES=LINES+8
      GO TO 500
191   WRITE(6,3012) ((GA(I,J),J=1,6),I=1,3)
      LINES=LINES+8
      GO TO 500
200   DELTM=FNTM-TEVN
      DO 210 I=1,6
      RI1(I)=RF1(I)
210   RI(I)=RF(I)
      IPR=IPRINT
      IPRINT=1
      ISPS=ISP2
      ISP2=NTP
      TRTM1=TEVN
      IF(ISOI1.EQ.1) GO TO 500
      CALL NTM(RI,RF,NTMC,-1)
      IPRINT=IPR
      ISP2=ISPS
      IF(ISPH.EQ.1) GO TO 220
      WRITE(6,3003)
      GO TO 500
220   TSOI1=DSI-DATEJ
      CALL PARTL(RSI,VSI,B1,BDT1,BDR1,PBT,PBR)
      DO 230 I=1,6
      EM(1,I)=PBT(I)
230   EM(2,I)=PBR(I)
      TSI=DSI-DATEJ
      RMSI=SQRT(RSI(1)*RSI(1)+RSI(2)*RSI(2)+RSI(3)*RSI(3))
      VMSI=SQRT(VSI(1)*VSI(1)+VSI(2)*VSI(2)+VSI(3)*VSI(3))
      WRITE(6,3042) TSI,RSI,RMSI,VSI,VMSI,B,BDT,BDR
      LINES=LINES+7
      WRITE(6,3041) ((EM(I,J),J=1,6),I=1,2)
      LINES=LINES+8
      IF(NQE.NE.0) GO TO 250
      DO 240 I = 1,6
240   RF1(I)=RF(I)
      GO TO 270
250   IPR=IPRINT
      IPRINT=1
      ISPS=ISP2
      ISP2=NTP
      CALL NTM(RI1,RF1,NTMC,-2)
      ISP2=ISPS
      IPRINT=IPR
      IF (ISPH.EQ.1) GO TO 270
      WRITE(6,3002)
```

```
      GO TO 500
270   TSI=DSI-DATEJ
      NO(1)=NTP
      CALL ORB(NTP,DSI)
      CALL EPHEM(1,DSI,1)
      DO 280 I=1,3
      RTPS(I)=RSI(I)+XP(I)*ALNGTH
280   RTPS(I+3)=VSI(I)+XP(I+3)*ALNGTH/TM
      IF (LINES.LT.MAX-7) GO TO 290
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
290   RMSI=SQRT(RSI(1)*RSI(1)+RSI(2)*RSI(2)+RSI(3)*RSI(3))
      VMSI=SQRT(VSI(1)*VSI(1)+VSI(2)*VSI(2)+VSI(3)*VSI(3))
      WRITE(6,3013) TSI,RSI,RMSI,VSI,VMSI,B,BDT,BDR
      LINES=LINES+13
      IF(IGP.EQ.3) GO TO 460
      DELTM=TSI-TEVN
      CALL PSIM(RI1,RTPS,ISTMC)
      IF(LINES.LT.MAX-9) GO TO 300
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
300   WRITE(6,3004) TSI,TEVN
      LINES=LINES+5
      DO 303 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 301
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
301   IF(NDIM.EQ.6) GO TO 302
      WRITE(6,3005) I
      LINES=LINES+1
302   WRITE(6,3006) (PSI(I,J),J=1,NDIM)
303   LINES=LINES+(NDIM-1)/6+1
      CALL PARTL(RSI,VSI,B1,BDT,BDR1,PBT,PBR)
      IF(LINES.LT.MAX-8) GO TO 310
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=10
310   WRITE(6,3014) PBT,PBR
      LINES=LINES+11
      DO 320 I=1,3
      A(1,I)=0.
      BB(1,I)=0.
      A(2,I)=0.
      BB(2,I)=0.
      DO 320 J=1,6
      A(1,I)=A(1,I)+PBT(J)*PSI(J,I)
      BB(1,I)=BB(1,I)+PBT(J)*PSI(J,I+3)
      A(2,I)=A(2,I)+PBR(J)*PSI(J,I)
320   BB(2,I)=BB(2,I)+PBR(J)*PSI(J,I+3)
      IF(LINES.LT.MAX-11) GO TO 330
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
330   WRITE(6,3015)((A(I,J),J=1,3),I=1,2),((BB(I,J),J=1,3),I=1,2)
      LINES=LINES+12
      DO 340 I=1,2
      DO 340 J=1,3
```

```
      ADA(I,J)=A(I,J)
340   ADA(I,J+3)=BB(I,J)
      DO 350 I=1,2
      DO 350 J=1,2
      PHI3(I,J)=0.
      DO 350 K=1,6
      DO 350 L=1,6
350   PHI3(I,J)=PHI3(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
      IF(LINES.LT.MAX-7) GO TO 360
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
360   WRITE(6,3016) ((PHI3(I,J),J=1,2),I=1,2)
      LINES=LINES+7
      IF(LINES.LT.MAX-13) GO TO 370
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
370   DO 371 I=1,2
      DO 371 J=1,2
371   DUM(I,J)=PHI3(I,J)
      CALL JACOBI(DUM,EGVL,DUM1,2,FOV)
      WRITE(6,3009) (I, EGVL(I),I=1,2)
      WRITE(6,3017) (I,(DUM1(I,J),J=1,2),I=1,2)
      LINES=LINES+13
      IF(IHYP1-2) 380,390,380
380   IF(LINES.LT.MAX-9) GO TO 381
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
381   CALL HYELS(1,PHI3,2)
      LINES=LINES+8
390   IF(IHYP1-1) 400,410,400
400   IF(LINES.LT.MAX-9) GO TO 401
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
401   CALL HYELS(3,PHI3,2)
      LINES=LINES+8
410   DO 420 I=1,2
      DO 420 J=1,2
      PHI3(I,J)=0.
      DO 420 K=1,3
420   PHI3(I,J)=PHI3(I,J)+BB(I,K)*BB(J,K)
      CALL MATIN(PHI3,PHI3,2)
      DO 430 I=1,3
      DO 430 J=1,2
      PHI2(I,J)=0.
      DO 430 K=1,2
430   PHI2(I,J)=PHI2(I,J)+BB(K,I)*PHI3(K,J)
      DO 440 I=1,3
      DO 440 J=1,3
      GA(I,J)=0.
      GA(I,J+3)=0.
      DO 440 K=1,2
      GA(I,J)=GA(I,J)-PHI2(I,K)*A(K,J)
440   GA(I,J+3)=GA(I,J+3)-PHI2(I,K)*BB(K,J)
      IF(LINES.LT.MAX-8) GO TO 450
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
```

```
      LINES=9
450   WRITE(6,3018) ((GA(I,J),J=1,6),I=1,3)
      LINES=LINES+8
      GO TO 500
460   CALL VARSIM(RI1,TEVN,TSI,ADA)
      IF(ISPH.EQ.0) GO TO 500
      GO TO 91
500   RETURN
3000  FORMAT(1H1//8X*SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIM
     $E *F8.3* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*)//)
3001  FORMAT(///8X*VEHICLE REACHED CLOSEST APPROACH ON MOST RECENT NOMIN
     $AL TRAJECTORY AT TRAJECTORY TIME*F12.3* DAYS*
     $                                          //53X*X*19X*Y*19X*Z*15X*RES
     $ULTANT*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X*VELOCITY
     $RELATIVE TO TARGET PLANET*4E20.10///1X130(1H*))
3002  FORMAT(8X*VEHICLE DID NOT REACH SPHERE OF INFLUENCE ON MOST RECENT
     $NOMINAL TRAJECTORY*/8X*RETURNING TO BASIC CYCLE*)
3003  FORMAT(8X*VEHICLE DID NOT REACH SPHERE OF INFLUENCE ON ORIGINAL NO
     $MINAL TRAJECTORY*/8X*RETURNING TO BASIC CYCLE*)
3004  FORMAT(///8X*STATE TRANSITION MATRIX -- PSI(*F8.3*,*F8.3*)*/)
3005  FORMAT(10X*ROW*I3)
3006  FORMAT(12X6E19.10)
3007  FORMAT(///8X*VARIATION MATRIX*/3(8X6E20.10/))
3008  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITIONS BEFORE CORRECTION*/
     $3(8X3E20.10/))
3009  FORMAT(//20X*EIGENVALUES OF ABOVE MATRIX*/3(22X,I2, E20.10/))
3010  FORMAT(//20X*EIGENVECTORS OF ABOVE MATRIX*/3(22X,I2,3E20.10/))
3011  FORMAT(///8X*GUIDANCE MATRIX -- FIXED TIME OF ARRIVAL GUIDANCE POL
     $ICY*/3(8X6E20.10/))
3012  FORMAT(///8X*GUIDANCE MATRIX -- THREE VARIABLE B-PLANE GUIDANCE PO
     $LICY*/3(8X6E20.10/))
3013  FORMAT(///8X*VEHICLE REACHED SPHERE OF INFLUENCE ON MOST RECENT NO
     $MINAL TRAJECTORY AT TRAJECTORY TIME*F10.3* DAYS*//53X*X*19X*Y*19X*
     $Z*15X*RESULTANT*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X
     $*VELOCITY RELATIVE TO TARGET PLANET*4E20.10//8X*B = *E20.10,5X
     $*B DOT T =*E20.10,5X*B DOT R = *E20.10///1X130(1H*))
3014  FORMAT(///8X*PARTIAL OF B DOT T WITH RESPECT TO STATE VECTOR*//
     $8X6E20.10///8X*PARTIAL OF B DOT R WITH RESPECT TO STATE VECTOR*//
     $8X6E20.10)
3015  FORMAT(///8X*GUIDANCE SUB-MATRIX A*/2(8X3E20.10/)//8X*GUIDANCE SUB
     $-MATRIX B*/2(8X3E20.10/))
3016  FORMAT(///8X*UNCERTAINTY IN TARGET CONDITION BEFORE CORRECTION*/
     $2(8X2E20.10/))
3017  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/2(22X,I2,2E20.10/))
3018  FORMAT(///8X*GUIDANCE MATRIX -- TWO VARIABLE B-PLANE GUIDANCE POLI
     $CY*/3(8X6E20.10/))
3040  FORMAT(8X*VEHICLE REACHED CLOSEST APPROACH ON ORIGINAL NOMINAL TRA
     $JECTORY AT TRAJECTORY TIME*F12.3* DAYS*//53X*X*19X*Y*19X*Z*15X*RES
     $ULTANT*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X*VELOCITY
     $RELATIVE TO TARGET PLANET*4E20.10)
3041  FORMAT(///8X*M MATRIX*//2(8X6E20.10/))
3042  FORMAT(8X*VEHICLE REACHED SPHERE OF INFLUENCE ON ORIGINAL NOMINAL
     $TRAJECTORY AT TRAJECTORY TIME*   F10.3* DAYS*//53X*X*19X*Y*19X*Z*
     $15X*RESULTANT*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X
     $*VELOCITY RELATIVE TO TARGET PLANET*4E20.10//8X*B = *E20.10,5X
     $*B DOT T =*E20.10,5X*B DOT R = *E20.10)
      END
```

```
      SUBROUTINE GUISIM(RI,TEVN,RI1)
      COMMON /CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RI1(6),RF(6),RF1(6),RI2(6),RF2(6),DUM(17),S(3,3),
     $P1(17,17),GA(3,6),EGVL(3),EGVCT(3,3),EXEC(3,3),DVE(3),
     $GAP(3,6),VEIG(9),ADA(3,6),DUM2(2,2),DX(17),DELX(17),DV(3),DVC(3)
      MAX=60
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
      DO 10 I=1,6
10    XF(I)=RF(I)
      IF (NQE.NE.0) GO TO 20
      DO 11 I=1,NDIM
11    XF1(I)=XF(I)
      DO 12 I=1,6
12    RF1(I)=RF(I)
      GO TO 30
20    CALL NTM(RI1,RF1,NTMC,2)
      DO 21 I=1,6
21    XF1(I)=RF1(I)
30    CALL PSIM(RI1,RF1,ISTMC)
      CALL DYNO(0)
      CALL NAVM(1,1)
      DO 50 I=1,6
50    RI2(I)=XI1(I)+ADEVX(I)
      CALL NTM(RI2,RF2,NTMC,3)
      DO 51 I=1,6
51    Z(I)=RF2(I)
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      WRITE(6,3001)
      LINES=12
      WRITE(6,3002) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
      LINES=LINES+NDIM
      WRITE(6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 33 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 31
```

```
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
31    IF(NDIM.EQ.6) GO TO 32
      WRITE(6,3013) I
      LINES=LINES+1
32    WRITE(6,3014) (PSI(I,J),J=1,NDIM)
33    LINES=LINES+(NDIM-1)/6+1
      IF(LINES.LT.MAX-8) GO TO 34
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
34    WRITE(6,3003)
      WRITE(6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      IF(LINES.LT.MAX-9) GO TO 35
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
35    WRITE(6,3005) TEVN,TRTM1
      LINES=LINES+5
      DO 38 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 36
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
36    IF(NDIM.EQ.6) GO TO 37
      WRITE(6,3013) I
      LINES=LINES+1
37    WRITE(6,3014) (P(I,J),J=1,NDIM)
38    LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(1)
      DO 60 I=1,6
60    ADEVX(I)=Z(I)+W(I)-XF1(I)
      DO 70 I=1,NDIM
      DUM(I)=0.
      DO 70 J=1,NDIM
70    DUM(I)=DUM(I)+PSI(I,J)*EDEVX(J)
      DO 71 I=1,NDIM
71    EDEVX(I)=DUM(I)
      ICODE2=1
80    ICODE=0
      K=0
      DO 81 J=1,3
      DO 81 I=1,3
      K=K+1
      S(I,J)=P(I,J)
81    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF(LINES.LT.MAX-16) GO TO 82
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
82    WRITE(6,1000)(I,EGVL(I),I=1,3)
      WRITE(6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
84    IF(IHYP1-2) 85,87,85
85    IF(LINES.LT.MAX-16) GO TO 86
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
```

```
      LINES=9
86    CALL HYELS(1,S,3)
      LINES=LINES+16
87    IF(IHYP1-1) 88,90,88
88    IF(LINES.LT.MAX-16) GO TO 89
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
89    CALL HYELS(3,S,3)
      LINES=LINES+16
90    IF (ICODE) 91,91,95
91    IF (IEIG) 95,95,92
92    K=0
      DO 93 J=1,3
      DO 93 I=1,3
      S(I,J)=P(I+3,J+3)
      K=K+1
93    VEIG(K)=S(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
      IF(LINES.LT.MAX-16) GO TO 94
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
94    WRITE(6,1003) (I,EGVL(I),I=1,3)
      WRITE(6,1004) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      ICODE=1
      GO TO 84
95    GO TO (100,120,170),ICODE2
100   IF (LINES.LT.MAX-NDIM-7) GO TO 53
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
53    WRITE(6,3008) (W(I),I=1,NDIM)
      LINES=LINES+NDIM+7
      IF (LINES.LT.MAX-NDIM-7) GO TO 72
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
72    WRITE(6,3010)(EDEVX(I),ADEVX(I),I=1,NDIM)
      LINES=LINES+NDIM+7
      DO 101 I=1,NDIM
      DO 101 J=1,NDIM
      P1(I,J)=P(I,J)
101   P(I,J)=PG(I,J)
      DO 102 I=1,6
102   RI1(I)=XG(I)
      DELTM=TEVN-TG
      TRTM1=TG
      CALL PSIM(RI1,RF1,ISTMC)
      IF (LINES.LT.MAX-9) GO TO 103
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
103   WRITE(6,3004) TEVN,TG
      LINES=LINES+5
      DO 106  I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 104
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
```

```
      LINES=9
104   IF(NDIM.EQ.6) GO TO 105
      WRITE(6,3013) I
      LINES=LINES+1
105   WRITE(6,3014) (PSI(I,J),J=1,NDIM)
106   LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(0)
      IF(LINES.LT.MAX-8) GO TO 107
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
107   WRITE(6,3003)
      WRITE(6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF (LINES.LT. MAX-9) GO TO 110
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
110   WRITE(6,3006) TEVN,TG
      LINES=LINES+5
      DO 113 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 111
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
111   IF(NDIM.EQ.6) GO TO 112
      WRITE(6,3013) I
      LINES=LINES+1
112   WRITE(6,3014) (P(I,J),J=1,NDIM)
113   LINES=LINES+(NDIM-1)/6+1
      ICODE2=2
      GO TO 80
120   NGE=NGE+1
      IGP=ICDT3(NGE)
      CALL GUIS(RF,RF1,IGP,TEVN,GA,ADA)
      IF(ISPH.EQ.0) GO TO 320
122   DO 123 I=1,3
      DO 123 J=1,6
      GAP(I,J)=0.
      DO 123 K=1,6
123   GAP(I,J)=GAP(I,J)+GA(I,K)*P(K,J)
      DO 124 I=1,3
      DO 124 J=1,3
      S(I,J)=0.
      DO 124 K=1,6
124   S(I,J)=S(I,J)+GAP(I,K)*GA(J,K)
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
      WRITE(6,3011) ((S(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      DO 125 I=1,3
      DO 125 J=1,3
125   EXEC(I,J)=S(I,J)
      ICODE2=1
119   CALL JACOBI(EXEC,EGVL,EGVCT,3,FOV)
      IF (LINES.LT.MAX-16) GO TO 126
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
```

```
      LINES=9
126   WRITE(6,2000) (I,EGVL(I),I=1,3)
      WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      IF(IGP.EQ.2.AND.ICODE2.EQ.1) GO TO 130
      IF (IHYP1.EQ.2) GO TO 128
      IF (LINES.LT.MAX-16) GO TO 127
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
127   CALL HYELS(1,S,3)
      LINES=LINES+16
128   IF(IHYP1.EQ.1) GO TO 130
      IF(LINES.LT.MAX-16) GO TO 129
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
129   CALL HYELS(3,S,3)
      LINES=LINES+16
      GO TO (130,160,200),ICODE2
130   DO 131 I=1,NDIM
      DX(I)=XF1(I)-XF(I)+ADEVX(I)
131   DELX(I)=XF1(I)-XF(I)+EDEVX(I)
      IF(LINES.LT.MAX-NDIM-6) GO TO 138
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
138   WRITE(6,3050) (DELX(I),DX(I),I=1,NDIM)
      LINES=LINES+NDIM+6
      DO 132 I=1,3
      DV(I)=0.
      DVC(I)=0.
      DO 132 J=1,6
      DV(I)=DV(I)+GA(I,J)*DX(J)
132   DVC(I)=DVC(I)+GA(I,J)*DELX(J)
      DO 133 I=1,3
133   DVE(I)=DV(I)-DVC(I)
      DVCM=SQRT(DVC(1)*DVC(1)+DVC(2)*DVC(2)+DVC(3)*DVC(3))
      IF (LINES.LT.MAX-20) GO TO 135
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
135   WRITE(6,3012) (DVC(I),DV(I),I=1,3),DVCM,(DVE(I),I=1,3)
      LINES=LINES+20
      X2=DVC(1)*DVC(1)
      Y2=DVC(2)*DVC(2)
      Z2=DVC(3)*DVC(3)
      DUM1=X2+Y2
      EXM=DUM1+Z2
      DUM3=SIGRES/EXM
      EXEC(1,1)=X2*(SIGPRO+DUM3)+Y2*EXM*SIGALP/DUM1+X2*Z2*SIGBET/DUM1
      EXEC(1,2)=DVC(1)*DVC(2)*(SIGPRO+DUM3-EXM*SIGALP/DUM1+Z2*SIGBET/
     $DUM1)
      EXEC(2,1)=EXEC(1,2)
      EXEC(1,3)=DVC(1)*DVC(3)*(SIGPRO+DUM3-SIGBET)
      EXEC(3,1)=EXEC(1,3)
      EXEC(2,2)=Y2*(SIGPRO+DUM3)+X2*EXM*SIGALP/DUM1+Y2*Z2*SIGBET/DUM1
      EXEC(2,3)=DVC(2)*DVC(3)*(SIGPRO+DUM3-SIGBET)
      EXEC(3,2)=EXEC(2,3)
      EXEC(3,3)=Z2*(SIGPRO+DUM3)+DUM1*SIGBET
```

```
      IF(LINES.LT.MAX-8) GO TO 136
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
136   WRITE(6,3015) ((EXEC(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      DO 140 I=1,NDIM
      DO 140 J=1,NDIM
140   P(I,J)=P1(I,J)
      DO 150 I=1,3
      DO 150 J=1,3
150   S(I,J)=EXEC(I,J)
      ICODE2=2
      GO TO 119
160   DO 161 I=1,3
      DO 161 J=1,3
161   P(I+3,J+3)=P(I+3,J+3)+S(I,J)
      DO 162 I=1,NDIM
      DO 162 J=1,NDIM
162   PG(I,J)=P(I,J)
      IF (LINES.LT.MAX-9) GO TO 163
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
163   WRITE(6,3016) TEVN,TEVN
      LINES=LINES+5
      DO 166 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 164
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
164   IF(NDIM.EQ.6) GO TO 165
      WRITE(6,3013) I
      LINES=LINES+1
165   WRITE (6,3014) (P(I,J),J=1,NDIM)
166   LINES=LINES+(NDIM-1)/6+1
      ICODE2=3
      GO TO 80
170   NN=3
      IF(IGP.EQ.2) NN=2
      DO 171 I=1,NN
      DO 171 J=1,NN
      S(I,J)=0.
      DO 172 K=1,6
      DO 172 L=1,6
172   S(I,J)=S(I,J)+ADA(I,K)*P(K,L)*ADA(J,L)
171   EXEC(I,J)=S(I,J)
      IF(LINES.LT.MAX-8) GO TO 180
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
180   IF(IGP.EQ.2) GO TO 190
      WRITE(6,3017) ((S(I,J),J=1,3),I=1,3)
      LINES=LINES+8
      ICODE2=3
      GO TO 119
190   WRITE(6,3018) ((S(I,J),J=1,2),I=1,2)
      LINES=LINES+8
      IF(LINES.LT.MAX-16) GO TO 191
      IPGN=IPGN+1
```

```
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
191   K=0
      DO 192 J=1,2
      DO 192 I=1,2
      K=K+1
192   VEIG(K)=S(I,J)
      CALL JACOBI(VEIG,EGVL,DUM2,2,FOV)
      WRITE(6,2000) (I,EGVL(I),I=1,2)
      WRITE(6,2002) (I,(DUM2(I,J),J=1,2),I=1,2)
      LINES=LINES+16
      IF(IHYP1.EQ.2) GO TO 194
      IF(LINES.LT.MAX-9) GO TO 193
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
193   CALL HYELS(1,S,2)
      LINES=LINES+8
194   IF(IHYP1.EQ.1) GO TO 200
      IF(LINES.LT.MAX-9) GO TO 195
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
195   CALL HYELS(3,S,2)
      LINES=LINES+8
200   DUM1=SQRT(DUM1)
      EXM=SQRT(EXM)
      AK1=ARES(NGE)
      S1 =APRO(NGE)
      AL1=AALP(NGE)
      BT1=ABET(NGE)
      DV(1)=DVC(1)*(S1+AK1/EXM)+(EXM*DVC(2)* AL1+DVC(1)*DVC(3)*BT1)/
     $DUM1
      DV(2)=DVC(2)*(S1+AK1/EXM)+(DVC(2)*DVC(3)*BT1- EXM*DVC(2)* AL1)/
     $DUM1
      DV(3)=DVC(3)*(S1+AK1/EXM)-BT1*DUM1
      DO 210 I=1,3
210   DVE(I)=DVC(I)+DV(I)
      IF (LINES.LT.MAX-8) GO TO 211
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
211   WRITE(6,3019) (DV(I),DVE(I),I=1,3)
      LINES=LINES+8
      DO 220 I=1,6
220   DX(I)=ADEVX(I)-EDEVX(I)
      NN=3
      IF (IGP.EQ.2) NN=2
      DO 230 I=1,NN
      DELX(I)=0.
      DO 230 J=1,6
230   DELX(I)=DELX(I)+ADA(I,J)*DX(J)
      DO 240 I=1,3
      RI1(I)=0.
240   RI1(I+3)=DV(I)
      DO 250 I=1,NN
      EGVL(I)=0.
      DO 250 J=1,6
250   EGVL(I)=EGVL(I)+ADA(I,J)*RI1(J)
      DO 260 I=1,NN
```

```
260   DV(I)=DELX(I)+EGVL(I)
      IF(LINES.LT.MAX-9 ) GO TO 270
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
270   WRITE(6,3020) (DELX(I),EGVL(I),I=1,NN)
      LINES=LINES+9
      IF (LINES.LT.MAX-8 ) GO TO 280
      IPGN=IPGN+1
      WRITE(6,3000) TEVN,IPROB,IPGN
      LINES=9
280   WRITE(6,3021) (DV(I),I=1,NN)
      LINES=LINES+8
      DO 290 I=1,6
290   XG(I)=XF1(I)
      DO 300 I=1,NDIM
      XI(I)=XF(I)
300   XI1(I)=XF1(I)
      TG=TEVN
      TRTM1=TEVN
      DO 310 I=1,3
      ADEVX(I+3)=ADEVX(I+3)+DVE(I)
310   EDEVX(I+3)=EDEVX(I+3)+DVC(I)
      GO TO 330
320   DO 321 I=1,NDIM
      DO 321 J=1,NDIM
321   P(I,J)=P1(I,J)
      DO 323 I=1,NDIM
      XI(I)=XF(I)
323   XI1(I)=XF1(I)
      TRTM1=TEVN
330   RETURN
3000  FORMAT(1H1//8X*SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIM
     $E *F8.3* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*))
3001  FORMAT(///8X*STATE VECTOR*//22X*ORIGINAL NOMINAL*7X*MOST RECENT NO
     $MINAL*13X*ACTUAL*)
3002  FORMAT(8XA10E20.10,5X,E20.10,5X,E20.10)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*STATE TRANSITION MATRIX -- PSI(*F8.3*,*F8.3*)*/)
3005  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT  -- P(*
     $F8.3*,*F8.3*)*/)
3008  FORMAT(///8X*ACTUAL DYNAMIC NOISE*//(8XE20.10))
3010  FORMAT(///8X*DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NO
     $MINAL TRAJECTORY*//15X*ESTIMATED*13X*ACTUAL*/(8X2E20.10))
3013  FORMAT(10X*ROW *I3)
3014  FORMAT(12X6E20.10)
1000  FORMAT(///20X*POSITION EIGENVALUES OF ABOVE MATRIX*/3(22XI2E20.10/
     $))
1001  FORMAT(///20X*POSITION EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.
     $10/))
1003  FORMAT(///20X*VELOCITY EIGENVALUES OF ABOVE MATRIX*/3(22XI2E20.10/
     $))
1004  FORMAT(///20X*VELOCITY EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.
     $10/))
2000  FORMAT(///20X*EIGENVALUES OF ABOVE MATRIX*/3(22XI2,E20.10/))
2001  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/3(22XI2,3E20.10/))
2002  FORMAT(///20X*EIGENVECTORS OF ABOVE MATRIX*/2(22XI2,2E20.10/))
3006  FORMAT(///8X*COVARIANCE MATRIX RELATING THE TIME OF THIS GUIDANCE
     $EVENT TO THAT AT THE LAST GUIDANCE EVENT -- P(*F8.3*,*F8.3*)*/)
3011  FORMAT(///8X*COVARIANCE MATRIX ASSOCIATED WITH VELOCITY COMPONENTS
```

```
     $*/3(8X3E20.10/))
3012 FORMAT(///8X*COMMANDED CORRECTION*9X*PERFECT CORRECTION*//3(8XE20.
     $10,5X,E20.10/)//10X*COMMANDED DELTA V. . .*E20.10
     $               ///8X*ERROR IN CORRECTION DUE TO NAVIGATION UNCERTAI
     $NTY*//3(8XE20.10/))
3015 FORMAT(///8X*EXECUTION ERROR MATRIX*//3(12X,3E20.10/))
3016 FORMAT(///8X*MODIFIED COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT
     $ -- P(*F8.3*,*F8.3*)*/)
3017 FORMAT(///8X*UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION*/
     $3(12X,3E20.10/))
3018 FORMAT(///8X*UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION*/
     $2(12X,2E20.10/))
3019 FORMAT(///8X*ACTUAL ERROR IN CORRECTION* 9X*ACTUAL CORRECTION*/
     $3(10XE20.10,10X,E20.10/))
3020 FORMAT(///8X*ERROR AT TARGET CONDITIONS*/8X*DUE TO NAVIGATION UNCE
     $RTAINTY* 4X*DUE TO EXECUTION ERROR*//(10XE20.10,10X,E20.10))
3021 FORMAT(///8X*ACTUAL ERROR AT TARGET AFTER CORRECTION*//(10XE20.10)
     $)
3050 FORMAT(///8X*DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL TRAJE
     $CTORY*//(15X*ESTIMATED*13X*ACTUAL*/(8X2E20.10))
      END
```

```
      SUBROUTINE HYELS(KS,P,N)
C
C
C     THIS SUBROUTINE COMPUTES AND PRINTS THE HYPERELLIPSOID
C     ASSOCIATED WITH THE MATRIX P AND SIGMA LEVEL KS
C
C     IFLAG IS A CODE WHICH IS DEFINED AS FOLLOWS
C        IFLAG = 0  PRINT XY HYPERELLIPSOID
C              = 1  PRINT XY, XZ, AND YZ HYPERELLIPSOIDS
C
C     THIS SUBROUTINE REQUIRES THE USE OF THE MATRIX INVERSION
C     ROUTINE -MATIN-
C
C
      DIMENSION P(3,3),PI(3,3),V(9)
      K2 =KS*KS
      K=0
      DO 10 J=1,N
      DO 10 I=1,N
      K=K+1
10    V(K)=P(I,J)
      CALL MATIN(V,V,N)
      K=0
      DO 20 J=1,N
      DO 20 I=1,N
      K=K+1
20    PI(I,J)=V(K)
      P12=2.*PI(1,2)
      IF(N.EQ.2) GO TO 30
      P13=2.*PI(1,3)
      P23=2.*PI(2,3)
      WRITE(6,100) KS
      WRITE(6,154)PI(1,1),PI(2,2),PI(3,3),P12,P13,P23,K2
      WRITE(6,500) PI(1,1),P12,PI(2,2),K2
   11 WRITE(6,501) PI(1,1) , P13, PI(3,3), K2
      WRITE(6,502) PI(2,2), P23, PI(3,3),K2
      GO TO 40
30    WRITE(6,100) KS
      WRITE(6,155) PI(1,1),P12,PI(2,2),K2
100   FORMAT(/30X*FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE *I1* SI
     $GMA LEVEL*/38X*THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION*//)
154   FORMAT(10XE10.3,8H X**2 + ,E10.3,8H Y**2 + ,E10.3,8H Z**2 + ,
     $E10.3,6H XY + ,E10.3,6H XZ + ,E10.3,6H YZ = I3//)
155   FORMAT(33XE10.3,8H X**2 + E10.3,6H XY + ,E10.3,8H Y**2 = I3//)
500   FORMAT(20X*XY HYPERELLIPSOID. . . . . . *E10.3,8H X**2 + ,E10.3,6H
     $ XY + ,E10.3,8H Y**2 = ,I3/)
501   FORMAT(20X*XZ HYPERELLIPSOID. . . . . . *E10.3,8H X**2 + ,E10.3,6H
     $ XZ + E10.3,8H Z**2 = ,I3/)
502   FORMAT(20X*YZ HYPERELLIPSOID. . . . . . *E10.3,8H Y**2 + ,E10.3,6H
     $ YZ + ,E10.3,8H Z**2 = ,I3/)
40    RETURN
      END
```

```
      SUBROUTINE HYPER(S,RP,VHL,GME,ELAT,A,E,XI,XL,XW,W,PV,Q,AZ,C3,P,
     1            DLA,RAL)
      DIMENSION W(3),PV(3),Q(3),B(3),S(3)
       PI=3.1415926536
       RAD=57.2957795
      C3=VHL**2
      CEL=COS(ELAT/RAD)
      AZ=AZ/RAD
      SAZ=SIN(AZ)
      E=1.+RP*C3/GME
      P=GME*(E**2-1.)/C3
      A=RP/(E-1.)
      SX=S(1)
      SY=S(2)
      SZ=S(3)
      DLA=ATAN(SZ/SQRT(SX**2+SY**2))*RAD
      IF(SX)30,31,30
   30 RAL = ATAN(SY/SX)
      IF(SX)32,31,33
   31 RAL = PI/2.
      IF(SY)32,33,33
   32 RAL = RAL + PI
   33 IF(RAL)34,35,35
   34 RAL = 2.*PI + RAL
   35 IF(ABS(DLA)-ELAT)10,10,11
   11 WZ=SQRT(1.-SZ**2)
      SAZ=SQRT((1.-SZ**2)/CEL**2)
      AZ=ATAN(SAZ/SQRT(1.-SAZ**2))
      WY=-WZ*SY*SZ/(SX**2+SY**2)
      GO TO 12
   10 WZ=CEL*SAZ
      WY=-(WZ*SY*SZ+SX*SQRT(1.-SZ**2-WZ**2))/(SX**2+SY**2)
   12 WX=-(WY*SY+WZ*SZ)/SX
      W(1)=WX
      W(2)=WY
      W(3)=WZ
      WM = SQRT(W(1)**2 +   W(2)**2 +W(3) **2)
      DO 40 I = 1,3
   40 W(I) = 1. * W(I)/WM
      B(1) = S(2)*W(3) - S(3)*W(2)
      B(2) = S(3)*W(1) - S(1)*W(3)
      B(3) = S(1)*W(2) - S(2)*W(1)
      BM = SQRT(B(1) **2 + B(2) **2 + B(3) **2)
      DO 45 I = 1,3
   45 B(I) = 1. * B(I)/BM
      CTAM=-1./E
      STAM=SQRT(1.-CTAM**2)
      DO 25 I=1,3
      PV(I)=S(I)*CTAM+B(I)*STAM
   25  Q(I)=S(I)*STAM-B(I)*CTAM
      XI=ATAN(SQRT(1.-WZ**2)/WZ)
      IF(-WY)50,51,50
   50 XL = ATAN(WX/(-WY))
      IF(-WY)52,51,53
   51 XL = PI/2.
      IF(WX)52,53,53
   52 XL = XL + PI
   53 IF(XL) 54,55,55
   54 XL = 2. * PI + XL
   55 IF(Q(3)) 60,61,60
```

```
60 XW = ATAN(PV(3)/Q(3))
   IF(Q(3))62,61,63
61 XW = PI/2.
   IF(PV(3))62,63,63
62 XW = XW + PI
63 IF(XW) 64,65,65
64 XW = 2. * PI + XW
65 XI = XI * RAD
   XL=XL*RAD
   XW=XW*RAD
   AZ=AZ*RAD
   RAL=RAL*RAD
   RETURN
   END
```

```
      SUBROUTINE HYPSV(R,P,E,C3,VHL,GME,RP,PV,Q,TA,XEQ,VEQ,VS,GAM,TS,
     1                 XEC,VEC,ECEQ)
      DIMENSION PV(3),Q(3),XEQ(3),VEQ(3),XEC(3),VEC(3),
     1          ECEQ(3,3),EQEC(3,3)
      PI = 3.1415926536
       RAD=57.2957795
      CTA=(P-R)/(E*R)
      STA=SQRT(1.-CTA**2)
      IF(CTA)40,41,40
   40 TA = ATAN(STA/CTA)
      IF(CTA)42,41,43
   41 TA = PI /2.
      IF(STA) 42,43,43
   42 TA = TA + PI
   43 IF(TA) 44,45,45
   44 TA = 2. * PI + TA
   45 VS=SQRT(C3+2.*GME/R)
      CGAM=SQRT(P*GME)/(VS*R)
      GAM=ATAN(SQRT(1.-CGAM**2)/CGAM)
      STAM=SIN(TA-GAM)
      CTAM=COS(TA-GAM)
      DO 20 I=1,3
      XEQ(I)=PV(I)*CTA*R+Q(I)*STA*R
   20 VEQ(I)=-PV(I)*STAM*VS+Q(I)*CTAM*VS
      DO 30 I=1,3
      DO 30 J=1,3
   30 EQEC(I,J)=ECEQ(J,I)
      DO 2 I = 1,3
      VEC(I) = 0.0
      XEC(I) = 0.0
      DO 2 K = 1,3
      VEC(I) = VEC(I) + EQEC(I,K)*VEQ(K)
    2 XEC(I) = XEC(I) + EQEC(I,K) *XEQ(K)
      CTS = 1./E*(P/R-1.)
      STS = SQRT(1.-CTS**2)
      DEN = RP/R *(1. + E)
      SF = SQRT(E**2 -1.) *STS/DEN
      F = ALOG(SF + SQRT(SF**2 + 1.))
      TS = GME/VHL**3*(E*SF-F)
      TA=TA*RAD
      GAM=GAM*RAD
      RETURN
      END
```

```
      SUBROUTINE INPUTZ(RS,NTP,IPRINT)
C
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR CONVERTING THE INPUT
C     INFORMATION INTO VARIABLES COMPATIBLE WITH THE REST OF THE
C     VIRTUAL MASS ROUTINES.
C
C
C
      DIMENSION IP(3)
      DIMENSION RS(6)
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON /COM/ NBODYI,   NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
C  INITIALIZATION OF VARIABLES
5000  INERR=0
      DO 2 I=1,44
      DO 2 J=1,4
2     F(I,J)=0.
      DO 3 I=1,4
3     IPRT(I)=I
      INC=1
      IPR=1
      IF(IPRINT.NE.0) GO TO 4
      CALL NEWPGE
      CALL SPACE(11)
4     V(2,1)=V(1,1)
      V(4,1)=V(3,1)
      CALL TIME(V(3,1),LYR,LMO,LDAY,LHR,LMIN,SECL,1)
      D2=V(3,1)+V(2,5) - V(1,1)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      TP = ALOG(V(4,5))
      V(3,6)=EXP(1.13756474179255+.509713741462307*TP+.14560181279278E-2
     **TP*TP)
      V(1,2)=RS(1)/V(1,6)
      V(1,3)=RS(2)/V(1,6)
      V(1,4)=RS(3)/V(1,6)
      V(3,2)=RS(4)/V(4,7)
      V(3,3)=RS(5)/V(4,7)
      V(3,4)=RS(6)/V(4,7)
60    DO 70 I=1,NBODYI
      IP = NO(I)
70    F(4*I-3,4)=PMASS(IP)
      ITRAT=3
      KOUNT= 1
      DO 80  I=2,4
      V(2,I)=V(1,I)
80    V(4,I)=V(3,I)
      NBODY=4*NBODYI-3
      V(13,3)=V(2,1)+V(3,5)
      IF(IPRINT.NE.0) GO TO 500
      WRITE(6,2000) V(1,6),V(1,5)
2000  FORMAT(//* UNITS . . . . . . .*E20.11* LNGTH/A.U.*5X,E20.11* TIME/
     1DAY*)
      WRITE(6,2001) NBODYI,(NO(I),I=1,NBODYI)
```

```
2001  FORMAT(I3* BODIES . . . . .*11I5)
     D=V(3,1)+2415020.
      WRITE(6,2002) MONTH(LMO),LDAY,LHR,LMIN,SECL,LYR,D
2002  FORMAT(20H LAUNCH DATE . . . .A10,I3,1H,I3,4H HR,I3,5H MIN,F7.3,5H
     * SEC,I5,21H. . . .JULIAN DATE. .F17.8)
     D=D2+2415020.
      WRITE(6,2003) MONTH(IMO),IDAY,IHR,IMIN,SECI,IYR,D
2003  FORMAT(20H ENCOUNTER DATE. . .A10,I3,1H,I3,4H HR,I3,5H MIN,F7.3,5H
     * SEC,I5,21H. . . .JULIAN DATE. .F17.8)
      WRITE(6,2004) V(1,1),V(4,5),V(3,6)
2004  FORMAT(* INITIAL TIME. . . .*E20.11/
     $* ACCURACY. . . . . . .*E20.11,10X*TRUE ANOMALY INCREMENT. . .*
     $E20.11)
500   RETURN
      END
```

```
      SUBROUTINE JACOBI (A,W2,V,N,FOD)
      DIMENSION  A(1), W2(1), V(1)
C
C   JACOBI METHOD FOR EIGENVALUES / EIGENVECTORS OF (A)(V) = (V)(-W2-).
C   THE (A) MATRIX SHOULD BE REAL, SYMMETRIC.
C   THRESHOLD VERSION OF JACOBI METHOD, PROGRESS FROM PIVOT ELEMENT
C    (IPIVOT,JPIVOT) TO ELEMENT (IPIVOT,JPIVOT+1) AFTER A PIVOT.
C   CODED BY  RL WOHLEN   MAY 1966.
C
C      SUBROUTINE ARGUMENTS
C   A  = INPUT  MATRIX TO BE DIAGONALIZED. SIZE(N,N). WILL BE DESTROYED.
C   W2 = OUTPUT VECTOR OF EIGENVALUES. (DIAGONAL OF DIAGONALIZED A).
C               SIZE(N).
C   V  = OUTPUT MATRIX OF EIGENVECTORS. SIZE(N,N).
C   N  = INPUT  SIZE OF A,W2,V.
C   FOD= INPUT  FINAL OFF-DIAGONAL ANNIHILATION VALUE
C
C
C
      KR = N
      KRP1 = KR + 1
C   SET INITIAL V MATRIX TO UNITY.
      II =-KR
      DO 10 I=1,N
      II = II + KRP1
      IJ = I - KR
      DO  5 J=1,N
      IJ = IJ + KR
    5 V(IJ) = 0.0
   10 V(II) = 1.0
      W2(1) = A(1)
      IF (N .EQ. 1)  RETURN
C
C   FIND LARGEST OFF-DIAGONAL ELEMENT FOR FIRST PIVOT.
      T1 = ABS(A(2))
      NM1 = N-1
      II =-KR
      DO 15 I=1,NM1
      II = II + KRP1
      IJ = II
      IP1 = I+1
      DO 15 J=IP1,N
      IJ = IJ + KR
   15 IF (ABS(A(IJ)) .GT. T1) T1=ABS(A(IJ))
      IF (T1 .LE. FOD) GO TO 60
C
C   SCAN UPPER OFF-DIAGONAL ELEMENTS OF MATRIX A BY ROWS UNTIL A VALUE
C   GREATER THAN T1 IS FOUND. PIVOT ON THIS ELEMENT (IP,JP).
   20 IREDO = 0
      IPIP =-KR
      DO 41 IP=1,NM1
      IPIP = IPIP + KRP1
      IPJP = IPIP
      JPIP = IPIP
      JPJP = IPIP
      IPP1 = IP+1
      DO 40 JP=IPP1,N
      IPJP = IPJP + KR
      JPIP = JPIP + 1
      JPJP = JPJP + KRP1
```

```
      IF (ABS(A(IPJP)) .LT. T1) GO TO 40
      IREDO = 1
C
C  COMPUTE ROTATION VALUES.
      DEL = A(IPIP) - A(JPJP)
      RAD = SQRT (DEL**2 + 4.*A(IPJP)**2)
      IF (DEL .GE. 0.) GO TO 24
      TN = (2. * A(IPJP)) / (DEL - RAD)
      GO TO 26
   24 TN = (2. * A(IPJP)) / (DEL + RAD)
   26 CS = 1. / SQRT (1. + TN**2)
      SN = TN * CS
C
C  COMPUTE EIGENVECTORS AND DIAGONALIZE MATRIX (A).
      AIPIP = A(IPIP)
      AJPJP = A(JPJP)
      AIPJP = A(IPJP)
      IIP = KR*(IP-1)
      IJP = KR*(JP-1)
      IPI = IP - KR
      JPI = JP - KR
      DO 35 I=1,N
      IIP = IIP + 1
      IJP = IJP + 1
      IPI = IPI + KR
      JPI = JPI + KR
      VIIP   = V(IIP)*CS + V(IJP)*SN
      V(IJP) =-V(IIP)*SN + V(IJP)*CS
      V(IIP) = VIIP
      AIIP   = A(IIP)*CS + A(IJP)*SN
      A(IJP) =-A(IIP)*SN + A(IJP)*CS
      A(JPI) = A(IJP)
      A(IIP) = AIIP
   35 A(IPI) = AIIP
      A(IPIP) = AIPIP*CS**2 + 2.*AIPJP*SN*CS + AJPJP*SN**2
      A(JPJP) = AIPIP*SN**2 - 2.*AIPJP*SN*CS + AJPJP*CS**2
      A(IPJP) = 0.0
      A(JPIP) = 0.0
   40 CONTINUE
   41 CONTINUE
      IF (IREDO .EQ. 1) GO TO 20
C
C  MAKE LARGEST OFF DIAGONAL ELEMENT OF (A) SMALLER THAN FOD.
      IF (T1 .LE. FOD) GO TO 60
    6 T1 = T1 * 1.E-3
      GO TO 20
C
C  PLACE DIAGONAL FROM A INTO W2 (EIGENVALUES).
   60 II =-KR
      DO 61 I=1,N
      II = II + KRP1
   61 W2(I) = A(II)
C
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LAMB(RL,RP,PSI,TF,GM,LOC,NTYS,A,E,P,VL,VP)
      FPB(AT)=2.*ATAN(AT/SQRT(1.-AT**2))
      PI=3.1415926536
      SNP=SIN(PSI)
      CSP=COS(PSI)
      C=SQRT(RL**2+RP**2-2.*RL*RP*CSP)
      AMIN=(RL+RP+C)/4.
      CSP=COS(PSI)
      S=2.*AMIN
  211 A=1.1*AMIN
  210 PM=SQRT(AMIN**3/GM)
      PERM=2.*PI*PM
      BETAM=FPB(SQRT((S-C)/S))
      TMIN=PM*(PI-BETAM+SIN(BETAM))
      TPMIN=PERM-TMIN
      P1=SQRT(2./GM)/3.
      TPP=P1*(S**1.5-(S-C)**1.5)
      TPPB=P1*(S**1.5+(S-C)**1.5)
      X=TMIN
      GO TO(11,12),NTYS
   11 IF(TF-TPP)13,13,16
   13 LOC=5
      WRITE(6,100)
  100 FORMAT( 6H HYPER)
      RETURN
   21 LOC=1
   23 A=AMIN
      WRITE(3,26)
   26 FORMAT(1X 6HA=AMIN)
      BETA=BETAM
      ALP=PI
      IT=0
      GO TO 30
   16 IF(TF-TMIN)17,21,18
   17 LOC=1
      GO TO 22
   18 LOC=2
      GO TO 22
   12 X=TPMIN
      IF(TF-TPPB)13,13,15
   15 IF(TF-TPMIN)19,21,50
   19 LOC=3
      GO TO 22
   50 LOC=4
   22 DO 40 IT=1,12
      ALP=FPB(SQRT(S/(2.*A)))
      SA=SIN(ALP)
      BETA=FPB(SQRT((S-C)/(2.*A)))
      SB=SIN(BETA)
      AG=SQRT(A**3/GM)
      P1=(ALP-SA)*AG
      P2=(BETA-SB)*AG
      AT=A
      AGM=1./SQRT(AT**3*GM)
      C2=AGM*S**2/SA
      C3=AGM*(S-C)**2/SB
      GO TO (1,2,3,4),LOC
    1 TFO=P1-P2
      C1=1.5*TFO/AT
      FPA=C1-C2+C3
```

```
      GO TO 10
    2 TFO=2.*PI*AG-P1-P2
      C1=1.5*TFO/AT
      FPA=C1+C2+C3
      GO TO 10
    3 TFO=P1+P2
      C1=1.5*TFO/AT
      FPA=C1-C2-C3
      GO TO 10
    4 TFO=2.*PI*AG-P1+P2
      C1=1.5*TFO/AT
      FPA=C1+C2-C3
   10 IF(ABS(TF-TFO)-TF/100000.)30,30,20
   20 HI=TF-TFO
      XKO=HI/FPA
      IF(ABS(XKO)-.5E-7)60,60,61
   61 A=A+XKO
      IF(A-AMIN)60,60,40
   60 FPA=   (X-TFO)/(AMIN-AT)
      A=AT+HI/FPA
   40 CONTINUE
      WRITE(3,101)TF,TFO,A,AMIN,TMIN,TPMIN
      LOC=6
      RETURN
  101 FORMAT(6F15.8)
   30 P=4.*A/C**2*(S-RL)*(S-RP)
      GO TO(51,52,52,51),LOC
   51 P1=(ALP+BETA)/2.
      GO TO 53
   52 P1=(ALP-BETA)/2.
   53 P=P*(SIN(P1))**2
      E=SQRT(1.-P/A)
      CVL=(P-RL)/(E*RL)
      SVL=(CVL*CSP-(P-RP)/(E*RP))/SNP
      IF(CVL)69,70,69
   69 VL=ATAN(SVL/CVL)
      IF(CVL)71,70,72
   70 VL=PI/2.
      IF(SVL)71,72,72
   71 VL=VL+PI
   72 IF(VL)73,74,74
   73 VL=VL+2.*PI
   74 VP=VL+PSI
      RETURN
      END
```

```
      SUBROUTINE MATIN(A,R,N)
      DIMENSION  A(1), R(1), IX(150), B(150), G(150), DETR(150)
C
C  MATRIX INVERSION (A**-1 = R) -- BORDERING METHOD.
C  THE DETERMINANT RATIO DET(I+1) / DET(I) IS PRINTED. DET(I) IS THE
C  DETERMINANT OF THE FIRST I BY I SUB-MATRIX OF A.
C  THE INVERSION CHECK R*A IS CALCULATED AND PRINTED.
C  MATRICES A AND R MAY SHARE SAME CORE LOCATIONS.(R*A CHECK IS INVALID)
C  THE MAXIMUM SIZES ARE
C     N = 150
C
C     SUBROUTINE ARGUMENTS
C  A  = INPUT  MATRIX TO BE INVERTED. SIZE(N,N).
C  R  = OUTPUT RESULT MATRIX. SIZE(N,N).
C  N  = INPUT  SIZE OF MATRIX A AND R. (MAX = 150)
C
 1001 FORMAT (26H1N EXCEEDS INV1 ALLOWABLE./ 17H0PROGRAM STOPPED.)
 1002 FORMAT (1H1,10X,18HMATRIX IS SINGULAR/11X,17HPROGRAM HAS ENDED)
 1902 FORMAT (1H1,10X,19HAMATRIX IS SINGULAR/11X,17HPROGRAM HAS ENDED)
C
      KR = N
      IF (N .LE. 150) GO TO 150
      WRITE(6,1001)
      STOP
C
  150 DO 160 I=2,N
      IX(I) = I
  160 CONTINUE
C     INVERT FIRST NON-ZERO ELEMENT IN FIRST COLUMN.
      DO 190 I=1,N
      IF (A(I) .NE. 0.) GO TO 220
  190 CONTINUE
      WRITE(6,1902)
      STOP
C
C     START INVERSION WITH ROW I.
  220 DETR(1) = A(I)
      R(I) = 1. / A(I)
      IF (N .EQ. 1) GO TO 999
C
      IX(I) = 1
      IX(1) = I
C     BORDERING LOOP.
      DO 630 L=2,N
          K = L
          L1 = L - 1
  250 S = 0.
                  MIXL = KR * (IX(L) - 1)
                  LL = IX(L) + MIXL
      DO 450 I=1,L1
                  MIXI = KR * (IX(I) - 1)
                  LI = IX(L) + MIXI
      B(I) = 0.
      G(I) = 0.
      DO 440 J=1,L1
                  MIXJ = KR * (IX(J) - 1)
                  IJ = IX(I) + MIXJ
                  JL = IX(J) + MIXL
      B(I) = B(I) - R(IJ)* A(JL)
                  JI = IX(J) + MIXI
```

```
                    LJ = IX(L) + MIXJ
      G(I) = G(I) - A(LJ)* R(JI)
  440 CONTINUE
      S = S + A(LI)* B(I)
  450 CONTINUE
      AL = A(LL)+ S
      IF (A(LL) .EQ. 0.) GO TO 480
      ALBAR = ABS (AL / A(LL))
      GO TO 490
  480 ALBAR = ABS (AL)
  490 IF (ALBAR .GE. .1E-6) GO TO 550
C
C     INTERCHANGE ROWS AND COLUMNS.
      K = K + 1
      IF (K .GT. N) GO TO 540
      IX L = IX(L)
      IX(L) = IX(K)
      IX(K) = IX L
      GO TO 250
  540 IF (ALBAR .GE. .1E-8) GO TO 550
  545 WRITE (6,1002)
      STOP
C
  550 R(LL)= 1. / AL
      DETR(L) = AL
      DO 570 I=1,L1
                    IL = IX(I) + MIXL
      LI = IX(L) + KR * (IX(I) - 1)
      R(IL)= B(I) * R(LL)
      R(LI)= G(I) * R(LL)
      DO 570 J=1,L1
      IJ = IX(I) + KR * (IX(J) - 1)
      R(IJ)= R(IJ)+ G(J) * R(IL)
  570 CONTINUE
  630 CONTINUE
C
C     COMPUTE INVERSION CHECK R*A.
      XOFF = 0.0
      DO 720 I=1,N
      DO 710 J=1,N
      X = 0.0
      KJA = KR * (J-1)
      DO 703 K=1,N
      IK = I + KR*(K-1)
      KJ = K + KJA
      X = X + R(IK) * A(KJ)
  703 CONTINUE
      IF (I .NE. J) GO TO 705
      G(I) = X
      GO TO 710
  705 IF (ABS(X) .LT. ABS(XOFF)) GO TO 710
      XOFF = X
      IOFF = I
      JOFF = J
  710 CONTINUE
  720 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE MENO(MMCODE,ICODE)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      REAL MNCN
      IF(ICODE.NE.0) GO TO 80
      DO 1 I=1,4
      DO 1 J=1,4
1     R(I,J)=0.
      IF(IMNF) 70,10,70
10    IF(MMCODE-9) 20,50,60
20    IF(MMCODE/2*2-MMCODE) 30,40,30
30    R(1,1)=MNCN(MMCODE+1)
      GO TO 70
40    R(1,1)=MNCN(MMCODE-1)
      R(2,2)=MNCN(MMCODE)
      GO TO 70
50    R(1,1)=MNCN(9)
      R(2,2)=MNCN(10)
      R(3,3)=MNCN(11)
      GO TO 70
60    R(1,1)=MNCN(12)
70    RETURN
80    IF(IAMNF.EQ.0) GO TO 140
      DO 81 I=1,4
      DO 81 J=1,4
81    AR(I,J)=0.
      IF(MMCODE-9) 90,120,130
90    IF(MMCODE/2*2-MMCODE) 100,110,100
100   AR(1,1)=AVARM(MMCODE+1)
      GO TO 70
110    AR(1,1)=AVARM(MMCODE-1)
      AR(2,2)=AVARM(MMCODE)
      GO TO 70
120   AR(1,1)=AVARM(9)
      AR(2,2)=AVARM(10)
      AR(3,3)=AVARM(11)
      GO TO 70
130   AR(1,1)=AVARM(12)
      GO TO 70
140   DO 141 I=1,4
      DO 141 J=1,4
141   AR(I,J)=R(I,J)
      GO TO 70
      END
```

```
      SUBROUTINE MUND(RI,RF,POSS)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION THETA(6,2),RI(6),RF(6),RPER(6)
      DO 1 I=1,6
      DO 1 J=1,2
1     THETA(I,J)=0.
      SAVE = PMASS(1)
      IPR=IPRINT
      IPRINT=1
      PMASS(1) = PMASS(1) + DELMUS*TM*TM/(ALNGTH*ALNGTH*ALNGTH)
      CALL NTM(RI,RPER,NTMC,0)
      J = 1
      DO 10 I = 1,6
   10 THETA(I,J) = (RPER(I) - RF(I))/DELMUS
      PMASS(1) = SAVE
      IF(POSS.GT.6.*SPHERE(NTP)*ALNGTH) GO TO 50
      SAVE = PMASS(NTP)
      PMASS(NTP) = PMASS(NTP)+DELMUP*TM*TM/(ALNGTH*ALNGTH*ALNGTH)
      CALL NTM(RI,RPER,NTMC,0)
      J = 2
      DO 15 I = 1,6
   15 THETA(I,J) = (RPER(I) - RF(I))/DELMUP
      PMASS(NTP) = SAVE
50    IF(IAUG.GT.3) GOTO 40
30    DO 31 I=1,6
      DO 31 J=1,2
31    PSI(I,J+6)=THETA(I,J)
      GO TO 100
40    IF(IAUG.EQ.9) GO TO 30
      DO 41 I=1,6
      DO 41 J=1,2
41    PSI(I,J+9)=THETA(I,J)
100   IPRINT=IPR
      RETURN
      END
```

```
      SUBROUTINE NAVM(N2,ICODE)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
C
C       INTERVAL TO OBTAIN THE NEW COVARIANCE MATRIX P(K+1,K+1)
C       FIRST, P(K+1,K) IS FOUND BY THE FORMULA
C
C         P(K+1,K) = PSI * P * PSI TRANSPOSE + Q
C
C       NEXT, P(K+1,K+1) IS COMPUTED BY
C
C         P(K+1,K+1) =P(K+1,K) - K*H*P(K+1,K)
C
C       INPUT DATA IS AS FOLLOWS
C         N1     --  DIMENSION OF P
C         N2     --  NUMBER OF ROWS IN H
C
C      THIS SUBROUTINE USES A MATRIX INVERSION ROUTINE  CALLED MATIN
C
C
      DIMENSION PPHT(4,4),A(16)
C     CALCULATE P PRIME
C     P PRIME = PSI * P * PSI TRANSPOSE  +  Q
      N1=NDIM
      DO 21 I = 1,N1
      DO 21 J = 1,N1
      PSIP(I,J) = 0.0
      DO 20 KK = 1,N1
      DO 20 L  = 1,N1
   20 PSIP(I,J)=PSIP(I,J)+PSI(I,L)* P(L,KK)*PSI(J,KK)
   21 PSIP(I,J) = PSIP(I,J) + Q(I,J)
      IF(ICODE.NE.0) GO TO 80
C     CALCULATE K
C     K = P PRIME * H TRANSPOSE * THE INVERSE OF (H* P PRIME *
C                   H TRANSPOSE  +  R)
      DO 31 I =1,N2
      DO 31 J =1,N2
      PPHT(I,J) = 0.0
      DO 30 K =1,N1
      DO 30 L =1,N1
   30 PPHT(I,J)= PPHT(I,J) +H(I,L)*PSIP(L,K)*H(J,K)
   31 PPHT(I,J) = PPHT(I,J) + R(I,J)
      DO 33 I=1,N2
      DO 33 J=1,N2
33    HPHR(I,J)=PPHT(I,J)
      IF(N2-1)41,42,41
   42 PPHT = 1./PPHT
      GO TO 51
   41 I = 1
      DO 40 J =1,N2
      DO 40 K =1,N2
      A(I) = PPHT(K,J)
   40 I = I + 1
      CALL MATIN(A,A,N2)
      I = 1
      DO 45 J = 1,N2
      DO 45 K = 1,N2
      PPHT(K,J) = A(I)
   45 I = I + 1
```

```
  900 FORMAT(4X,4(F10.3,5X))
   51 DO 50 I = 1,N1
      DO 50 J = 1,N2
      AK(I,J) = 0.0
      DO 50 K = 1,N2
      DO 50 L = 1,N1
   50 AK (I,J)  =  AK (I,J) +PSIP(I,L)*H(K,L)*PPHT(K,J)
C     CALCULATE P   P= PP - K*H*PP
      DO 55 I = 1,N1
      DO 55 J = 1,N1
      P   (I,J)=0.0
      DO 60 K = 1, N1
      DO 60 L = 1, N2
   60 P(I,J) = P(I,J) + AK(I,L)*H(L,K)*PSIP(K,J)
   55 P(I,J) = PSIP(I,J) - P(I,J)
C     MAKE P A SYMMETRIC MATRIX BY USING THE MEAN OF THE CORRESPONDING
C     LOWER AND UPPER OFF DIAGONAL ELEMENTS
65    NM = N1- 1
      DO 70 I = 1,NM
      L = I + 1
      DO 70 J = L,N1
      P(I,J) = 0.5*(P(I,J)+P(J,I))
   70 P(J,I) = P(I,J)
      GO TO 100
80    DO 90 I=1,NDIM
      DO 90 J=1,NDIM
90    P(I,J)=PSIP(I,J)
      GO TO 65
100   RETURN
      END
```

```
      SUBROUTINE NDTM(RI,RF)
C THIS SUBROUTINE CALCULATES THE STATE TRANSITION MATRIX USING
C NUMERICAL DIFFERENCING
C
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION T(6),U(6),RI(6),RF(6),RP(6)
      DO 1 I=1,6
1     RP(I)=RF(I)
      SAVE=ACC
      IPR=IPRINT
      IPRINT=1
      IF(NDACC.EQ.0) GO TO 5
      ACC=ACCND
      CALL NTM(RI,RP,2,0)
5     GO TO (10,20), NTMC
10    IF(NDACC.NE.0) GO TO 20
      CALL NTM(RI,RP,2,0)
20    N=1
      F1=FACP
      F2=FACV
      IF(DELTM.GE.10.) GO TO 21
      FACP=10.*FACP
      FACV=10.*FACV
      GO TO 60
21    IF(DELTM.LE.100.) GO TO 60
      FACP=.1*FACP
      FACV=.1*FACV
      GO TO 60
   60 DO 30 II=1,6
30    T(II)=RI(II)
      IF(N-4) 35,40,40
   35 T(N) =RI(N)+FACP
      GO TO 41
   40 T(N) =RI(N)+FACV
41    CALL NTM(T,U,2,0)
      DO 45 M = 1,6
      IF(N - 4) 50,55,55
50    PSI(M,N)=(U(M)-RP(M))/FACP
      GO TO 45
55    PSI(M,N)=(U(M)-RP(M))/FACV
   45 CONTINUE
      N = N + 1
      IF(N -6) 60,60,65
65    ACC=SAVE
      IPRINT=IPR
      FACP=F1
      FACV=F2
      RETURN
      END
```

```
      SUBROUTINE NEWPGE
      COMMON /COM/V(16,7),F(4 .. ..,, ......
      COMMON/COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      IPG=IPG+1
      WRITE (6,1)
    1 FORMAT                                                        (120H
     $1    V I R T U A L    M A S S    P R O G R A M    F O R    C O M P
     $U T I N G    S P A C E    T R A J E C T O R I E S       )
C
      WRITE (6,2) KL,IPG
    2 FORMAT (90X8HPROBLEM I5,6X5HPAGE I4///)
      LINCT=5
C     WHEN IPG=1,ONLY TITLE,PROBLEM NUMBER,AND PAGE NUMBER ARE GIVEN, AS
C     THIS SIGNALS INPUT DIAGNOSTICS ARE TO BE GIVEN, OR INPUT DATA
C     IS TO BE LISTED.
C
      IF(IPG.EQ.1) GO TO 10
      WRITE (6,3)
3      FORMAT(                          40X80H           X - COMP.
     $    Y - COMP.              Z - COMP.           RESULTANT    )
      LINCT=6
10    RETURN
      END
```

```
      SUBROUTINE NJEXN(JC3,JINJT,NDD,NTT,DDJD,TTJD,HHR1,HHV1,S)
      DIMENSION NTDD(5),NTTT(5),DDN(3),TTN(3),DDR1(3),DDV1(3),TTR7(3),
     1 TTV7(3),CCN(3),CCR1(3),CCV1(3),
     2  HHV2(3),ECEQ(3,3),HHVEQ(3),HHN(3),HHPV(3),HHQ(3),
     3 HHRQ1(3),HHRQ2(3),HHVQ1(3),HHVQ2(3),HHR1(3),HHR2(3),HHV1(3)
      DIMENSION XL(6),SXL(6),S(3)
      DIMENSION EQEC(3,3)
      COMMON /BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      CALL        CONST(MODE,NDD,NTT,PI,RAD,AU,AUDAY,AUS,CONV,SSG,PR,HHTA
     1 ,ANG1,ANG2,TIM1,TIM2,DDLAT,DDLON,DDIQ,DDLQ,ROT)
      CALL ORB(NDD,DDJD)
      NO(1)=NDD
      CALL EPHEM(1,DDJD,1)
      DO 5 I=1,6
    5 XL(I)=XP(I)
      DO 6 I=1,3
    6 DDV1(I)=XL(I+3)
      DDG=PMASS(NDD)*(AUS**3/86400.**2)
      DDS=SPHERE(NDD)*AUS
      CALL TIME(DDJD,NTDD(1),NTDD(2),NTDD(3),NTDD(4),NTDD(5),SDD,1)
      CALL TIME(TTJD,NTTT(1),NTTT(2),NTTT(3),NTTT(4),NTTT(5),STT,1)
      CL=COS(DDLQ)
      SL=SIN(DDLQ)
      CI=COS(DDIQ)
      SI=SIN(DDIQ)
      ECEQ(1,1)=CL
      ECEQ(1,2)=SL
      ECEQ(1,3)=0.
      ECEQ(2,1)=-SL*CI
      ECEQ(2,2)=CL*CI
      ECEQ(2,3)=SI
      ECEQ(3,1)=SL*SI
      ECEQ(3,2)=-CL*SI
      ECEQ(3,3)=CI
      CALL ORB(NTT,TTJD)
      NO(1)=NTT
      CALL EPHEM(1,TTJD,1)
      RL=SQRT(XL(1)*XL(1)+XL(2)*XL(2)+XL(3)*XL(3))
      RP=SQRT(XP(1)*XP(1)+XP(2)*XP(2)+XP(3)*XP(3))
      VL=SQRT(XL(4)*XL(4)+XL(5)*XL(5)+XL(6)*XL(6))
      VP=SQRT(XP(4)*XP(4)+XP(5)*XP(5)+XP(6)*XP(6))
      TF=TTJD-DDJD
      CALL PLANE(XL,XP,CCPSI,CCI,CCL,CCN,NTYS)
      CALL LAMB(RL,RP,CCPSI,TF,SSG,LOC,NTYS,CCA,CCE,CCP,CCTA1,CCTA7)
      IF(LOC-5)10,1,1
   10 CCW=2.*PI-CCTA1
      IF(XP(3)*(PI-CCPSI))20,20,30
   20 CCW=PI-CCTA1
   30 CONTINUE
      IVHL = -1
      SRL = RL
      STA1 = CCTA1
      RL = RP
      CCTA1 = CCTA7
      DO 29 I = 1,3
      SXL(I+3) = XL(I+3)
   29 XL(I+3) = XP(I+3)
      SJC3 = JC3
      JC3 = 1
```

```
   31 IVHL = IVHL + 1
C            MEAN ROUTINE
      P1=(CCA-RL)/(CCA*CCE)
      P2=CCA*SQRT(1.-CCE*CCE)
      P2=RL*SIN(CCTA1)/P2
      IF(P1)40,41,40
   40 CCEA1=ATAN(P2/P1)
      IF(P1)42,41,43
   41 CCEA1=PI/2.
      IF(P2)42,43,43
   42 CCEA1=CCEA1+PI
   43 IF(CCEA1)44,46,46
   44 CCEA1=CCEA1+2.*PI
   46 CCM1=CCEA1-CCE*P2
C
      CALL POSVL(CCA,CCE,CCI,CCL,CCW,CCM1,CCN,CCR1,CCRM1,CCV1,CCVM1,SSG)
      DO 50 I=1,3
   50 HHV2(I)=CONV*CCV1(I)-CONV*XL(I+3)
      VHL=SQRT(HHV2(1)**2+HHV2(2)**2+HHV2(3)**2)
      IF(JC3)33,32,33
   32 VHL=SQRT(VHL*VHL-2.*DDG/DDS)
   33 CONTINUE
      IF(HHV2(1))60,61,60
   60 HHVRA=ATAN(HHV2(2)/HHV2(1))
      IF(HHV2(1))62,61,63
   61 HHVRA=PI/2.
      IF(HHV2(2))62,63,63
   62 HHVRA=HHVRA+PI
   63 IF(HHVRA)64,65,65
   64 HHVRA=HHVRA+2.*PI
   65 HHVRA=HHVRA*RAD
      HHVDC=ATAN(HHV2(3)/VHL)*RAD
      IF(IVHL)70,70,75
   70 VHP = VHL
      DO 72 I=1,3
   72 S(I)=HHV2(I)
      RAP = HHVRA
      DPA = HHVDC
      RL = SRL
      CCTA1 = STA1
      DO 71 I = 1,3
   71 XL(I+3) = SXL(I+3)
      JC3 = SJC3
      GO TO 31
   75 CONTINUE
      HHVEQ(1)=ECEQ(1,1)*HHV2(1)+ECEQ(1,2)*HHV2(2)+ECEQ(1,3)*HHV2(3)
      HHVEQ(2)=ECEQ(2,1)*HHV2(1)+ECEQ(2,2)*HHV2(2)+ECEQ(2,3)*HHV2(3)
      HHVEQ(3)=ECEQ(3,1)*HHV2(1)+ECEQ(3,2)*HHV2(2)+ECEQ(3,3)*HHV2(3)
      HHVEQ(1)=HHVEQ(1)/VHL
      HHVEQ(2)=HHVEQ(2)/VHL
      HHVEQ(3)=HHVEQ(3)/VHL
      DDAZ = 90.0
   35 CALL HYPER(HHVEQ,PR,VHL   ,DDG,DDLAT,HHA,HHE,HHI,HHL,HHW,HHN,HHPV,H
     1HQ,DDAZ,C3,HHP,DLAQ,RALQ)
      RJ=HHP/(1.+HHE*COS(HHTA/RAD))
      CALL HYPSV(RJ ,HHP,HHE,C3,VHL   ,DDG,PR,HHPV,HHQ,HHTA1,HHRQ1,HHVQ1,
     1 HHVM1,PTH   ,HHT1,HHR1,HHV1,ECEQ)
      CALL AUX(HHN,DDLAT,DDLON,DDAZ,HHPV,HHQ,HHTA,ANG1,ANG2,TIM1,TIM2,HH
     1VEQ,HHE,PR,DDG,ROT,DDJD,TL,TB,PHI,THI,RAI,AZI,TINJ,TC)
      IF(JINJT)1,45,1
```

```
 1 CONTINUE
   CALL OT2(XL,XP,RL,CCVM1,CCPSI,CCA,CCI,VL,RP,VP,                    CCT
  1A1,CCTA7,TL,TINJ,NTDD,TF,NTTT,C3,VHL   ,DLAQ,RALQ,RJ,HHVM1,PTH,VHP,
  2DPA,RAP,HHE,DDAZ,TB,PHI,THI,RAI,AZI,TC,CCE)
38 DDJD = DDJD + TINJ/24.
45 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE NTM(RI,RF,NTC,ICODE)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR GENERATING THE NOMINAL
C     TRAJECTORY WHICH IS USED IN THE VARIOUS MODES OF OPERATION
C     IN THE STEAP PROGRAM.
C
C     THE INPUT ARGUMENTS ARE DESCRIBED AS FOLLOWS.
C        RI     --  INITIAL POSITION AND VELOCITY OF VEHICLE IN
C                   HELIOCENTRIC ECLIPTIC COORDINATES
C        NTC    --  NOMINAL TAJECTORY CODE
C                   =1  --  PATCHED CONIC TRAJECTORY
C                   =2  --  VIRTUAL MASS TRAJECTORY
C
C     OUTPUT
C     THE OUTPUT ARGUMENT IS
C        RF     --  THE FINAL POSITION AND VELOCITY OF THE VEHICLE IN
C                   HELIOCENTRIC ELIPTIC COORDINATES
C
C
      DIMENSION RI(6),RF(6)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION NBS(11)
      D1=TRTM1+DATEJ
      IF(ABS(ICODE).EQ.3) GO TO 60
5     GO TO (10,20),NTC
10    WRITE(6,1000)
1000  FORMAT(///8X*PATCHED CONIC TRAJECTORY IS NOT AVAILABLE IN THIS DEC
     $K*)
      GO TO 50
20    IF(ICODE.NE.0) GO TO 21
      ISPH=1
      ICL=1
      GO TO 30
21    IF(ICODE.LT.0) GO TO 34
      IF(ICODE-2) 22,26,30
22    IF(ISOI1.EQ.0) GO TO 23
      ISPH=1
      GO TO 24
23    ISPH=0
24    IF(ICA1.EQ.0) GO TO 25
      ICL=1
      GO TO 35
```

```
25    ICL=0
      GO TO 35
26    IF(ISOI2.EQ.0) GO TO 27
      ISPH=1
      GO TO 28
27    ISPH=0
28    IF(ICA2.EQ.0) GO TO 29
      ICL=1
      GO TO 35
29    ICL=0
      GO TO 35
30    IF(ISOI3.EQ.0) GO TO 31
      ISPH=1
      GO TO 32
31    ISPH=0
32    IF(ICA3.EQ.0) GO TO 33
      ICL=1
      GO TO 35
33    ICL=0
      GO TO 35
34    ISPH=0
      ICL=0
35    CALL VMP(RI,ACC,D1,TRTM1,DELTM,RF,ISP2)
      IF(ICODE.LE.0) GO TO 50
      IF(ICODE-2) 36,41,45
36    IF(ISPH.EQ.0) GO TO 38
      IF(ISOI1.EQ.1) GO TO 38
      ISOI1=1
      DO 37 I=1,3
      RSOI1(I)=RSI(I)
37    VSOI1(I)=VSI(I)
      TSOI1=DSI-DATEJ
      BSI1=B
      BDTSI1=BDT
      BDRSI1=BDR
      RMP=SQRT(RSOI1(1)*RSOI1(1)+RSOI1(2)*RSOI1(2)+RSOI1(3)*RSOI1(3))
      VMP=SQRT(VSOI1(1)*VSOI1(1)+VSOI1(2)*VSOI1(2)+VSOI1(3)*VSOI1(3))
      IPGN=IPGN+1
      WRITE(6,2000) IPROB,IPGN,TSOI1,RSOI1,RMP,VSOI1,VMP,BSI1,BDTSI1,
     $BDRSI1
2000  FORMAT(1H1//100X*PROBLEM*I10,5X,*PAGE*I5////////1X,130(1H*)///
     $8X*ORIGINAL NOMINAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT
     $TRAJECTORY TIME *F10.5* DAYS*///53X*X*19X*Y*19X*Z*15X*RESULTANT*/
     $8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/
     $8X*VELOCITY RELATIVE TO TARGET PLANET*4E20.10
     $///8X*B = *E20.10,5X*B DOT T = *E20.10,5X*B DOT R = *E20.10///
     $1X130(1H*))
38    IF(ICA1.EQ.1) GO TO 40
      DO 39 I=1,6
39    RCA1(I)=RC(I)
      TCA1=DC-DATEJ
      IF(ICL.EQ.0) GO TO 40
      ICA1=1
      RMP=SQRT(RCA1(1)*RCA1(1)+RCA1(2)*RCA1(2)+RCA1(3)*RCA1(3))
      VMP=SQRT(RCA1(4)*RCA1(4)+RCA1(5)*RCA1(5)+RCA1(6)*RCA1(6))
      IPGN=IPGN+1
      WRITE(6,2001) IPROB,IPGN,TCA1,(RCA1(I),I=1,3),RMP,(RCA1(I),I=4,6),
     $VMP
2001  FORMAT(1H1//100X*PROBLEM*I10,5X*PAGE*I5////////1X130(1H*)///8X
     $*ORIGINAL NOMINAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO
```

```
     $TARGET PLANET AT *F10.5* DAYS*    ///53X*X*19X*Y*19X*Z*15X*RESULTAN
     $T*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X*VELOCITY RELAT
     $IVE TO TARGET PLANET*4E20.10///1X130(1H*))
40    IF(ITR.NE.4) GO TO 50
      IF(NQE.NE.0) GO TO 50
41    IF(ISPH.EQ.0) GO TO 43
      IF(ISOI2.EQ.1) GO TO 43
      ISOI2=1
      DO 42 I=1,3
      RSOI2(I)=RSI(I)
42    VSOI2(I)=VSI(I)
      TSOI2=DSI-DATEJ
      BSI2=B
      BDTSI2=BDT
      BDRSI2=BDR
      RMP=SQRT(RSOI2(1)*RSOI2(1)+RSOI2(2)*RSOI2(2)+RSOI2(3)*RSOI2(3))
      VMP=SQRT(VSOI2(1)*VSOI2(1)+VSOI2(2)*VSOI2(2)+VSOI2(3)*VSOI2(3))
      IPGN=IPGN+1
      WRITE(6,2002) IPROB,IPGN,TSOI2,RSOI2,RMP,VSOI2,VMP,BSI2,BDTSI2,
     $BDRSI2
2002  FORMAT(1H1//100X*PROBLEM*I10,5X,*PAGE*I5////////1X,130(1H*)///
     $8X*MOST RECENT NOMINAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE
     $AT TRAJECTORY TIME *F10.5* DAYS*///53X*X*19X*Y*19X*Z*15X*RESULTANT
     $*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/
     $8X*VELOCITY RELATIVE TO TARGET PLANET*4E20.10
     $///8X*B = *E20.10,5X*B DOT T = *E20.10,5X*B DOT R = *E20.10///
     $1X130(1H*))
43    IF(ICA2.EQ.1) GO TO 50
      DO 44 I=1,6
44    RCA2(I)=RC(I)
      TCA2=DC-DATEJ
      IF(ICL.EQ.0) GO TO 50
      ICA2=1
      RMP=SQRT(RCA2(1)*RCA2(1)+RCA2(2)*RCA2(2)+RCA2(3)*RCA2(3))
      VMP=SQRT(RCA2(4)*RCA2(4)+RCA2(5)*RCA2(5)+RCA2(6)*RCA2(6))
      IPGN=IPGN+1
      WRITE(6,2003) IPROB,IPGN,TCA2,(RCA2(I),I=1,3),RMP,(RCA2(I),I=4,6),
     $VMP
2003  FORMAT(1H1//100X*PROBLEM*I10,5X*PAGE*I5////////1X130(1H*)///8X
     $*MOST RECENT NOMINAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH
     $TO TARGET PLANET AT *F10.3* DAYS*///53X*X*19X*Y*19X*Z*15X*RESULTAN
     $T*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X*VELOCITY RELAT
     $IVE TO TARGET PLANET*4E20.10///1X130(1H*))
      GO TO 50
45    IF(ISPH.EQ.0) GO TO 47
      IF(ISOI3.EQ.1) GO TO 47
      ISOI3=1
      DO  46 I=1,3
      RSOI3(I)=RSI(I)
46    VSOI3(I)=VSI(I)
      TSOI3=DSI-DATEJ
      BSI3=B
      BDTSI3=BDT
      BDRSI3=BDR
      RMP=SQRT(RSOI3(1)*RSOI3(1)+RSOI3(2)*RSOI3(2)+RSOI3(3)*RSOI3(3))
      VMP=SQRT(VSOI3(1)*VSOI3(1)+VSOI3(2)*VSOI3(2)+VSOI3(3)*VSOI3(3))
      IPGN=IPGN+1
      WRITE(6,2004) IPROB,IPGN,TSOI3,RSOI3,RMP,VSOI3,VMP,BSI3,BDTSI3,
     $BDRSI3
2004  FORMAT(1H1//100X*PROBLEM*I10,5X,*PAGE*I5////////1X,130(1H*)///
```

```
     $8X*ACTUAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT TRAJECTORY
     $ TIME *           F10.5* DAYS*///53X*X*19X*Y*19X*Z*15X*RESULTANT*/
     $8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/
     $8X*VELOCITY RELATIVE TO TARGET PLANET*4E20.10
     $///8X*B = *E20.10,5X*B DOT T = *E20.10,5X*B DOT R = *E20.10///
     $1X130(1H*))
47    IF(ICA3.EQ.1) GO TO 50
      DO 48 I=1,6
48    RCA3(I)=RC(I)
      TCA3=DC-DATEJ
      IF(ICL.EQ.0) GO TO 50
      ICA3=1
      RMP=SQRT(RCA3(1)*RCA3(1)+RCA3(2)*RCA3(2)+RCA3(3)*RCA3(3))
      VMP=SQRT(RCA3(4)*RCA3(4)+RCA3(5)*RCA3(5)+RCA3(6)*RCA3(6))
      IPGN=IPGN+1
      WRITE(6,2005) IPROB,IPGN,TCA3,(RCA3(I),I=1,3),RMP,(RCA3(I),I=4,6),
     $VMP
2005  FORMAT(1H1//100X*PROBLEM*I10,5X*PAGE*I15///////1X130(1H*)///8X
     $*ACTUAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO TARGET PLA
     $NET AT *          F10.5* DAYS*   ///53X*X*19X*Y*19X*Z*15X*RESULTAN
     $T*/8X*POSITION RELATIVE TO TARGET PLANET*4E20.10/8X*VELOCITY RELAT
     $IVE TO TARGET PLANET*4E20.10///1X130(1H*))
50    IF(ABS(ICODE).NE.3) GO TO 100
      NBOD=NBODS
      ACC=ACCS
      DO 51 I=1,NBOD
51    NB(I)=NBS(I)
      PMASS(1)=SAVE1
      PMASS(NTP)=SAVE2
      SMJR(K1+1)=SAVE3
      SMJR(K1+2)=SAVE4
      IF(NTP.GT.5) GO TO 80
      CN(K2+1)=SAVE5
      CN(K2+2)=SAVE6
      CN(K2+3)=SAVE7
      CN(K2+4)=SAVE8
      CN(K3+1)=SAVE10
      CN(K3+2)=SAVE11
      CN(K3+3)=SAVE12
      CN(K3+4)=SAVE13
      GO TO 100
80    ST(K2+1)=SAVE5
      ST(K2+2)=SAVE6
      ST(K3+1)=SAVE10
      ST(K3+2)=SAVE11
      GO TO 100
60    DO 61 I=1,NBOD
61    NBS(I)=NB(I)
      NBODS=NBOD
      NBOD=NBOD1
      DO 62 I=1,NBOD
62    NB(I)=NB1(I)
      ACCS=ACC
      ACC=ACC1
      SAVE1=PMASS(1)
      PMASS(1)=PMASS(1)+DMUSB*TM*TM/(ALNGTH*ALNGTH*ALNGTH)
      SAVE2=PMASS(NTP)
      PMASS(NTP)=PMASS(NTP)+DMUPB*TM*TM/(ALNGTH*ALNGTH*ALNGTH)
      K1=2*(NTP-2)
      SAVE3=SMJR(K1+1)
```

```
      SAVE4=SMJR(K1+2)
      SMJR(K1+1)=SMJR(K1+1) + DAB/ALNGTH
      SMJR(K1+2)=SMJR(K1+2) + DAB/ALNGTH
      IF(NTP.GT.5) GO TO 70
      K2=20*NTP-28
      SAVE5=CN(K2+1)
      SAVE6=CN(K2+2)
      SAVE7=CN(K2+3)
      SAVE8=CN(K2+4)
      CN(K2+1) = CN(K2+1) + DEB
      CN(K2+2) = CN(K2+2) + DEB
      CN(K2+3) = CN(K2+3) + DEB
      CN(K2+4) = CN(K2+4) + DEB
      K3=20*(NTP-2)
      SAVE10=CN(K3+1)
      SAVE11=CN(K3+2)
      SAVE12=CN(K3+3)
      SAVE13=CN(K3+4)
      CN(K3+1)=CN(K3+1)+DIB
      CN(K3+2)=CN(K3+2)+DIB
      CN(K3+3)=CN(K3+3)+DIB
      CN(K3+4)=CN(K3+4)+DIB
      GO TO 5
70    K2=10*NTP-14
      SAVE5=ST(K2+1)
      SAVE6=ST(K2+2)
      ST(K2+1) = ST(K2+1) + DEB
      ST(K2+2) = ST(K2+2) + DEB
      K3=10*(NTP-2)
      SAVE10=ST(K3+1)
      SAVE11=ST(K3+2)
      ST(K3+1)=ST(K3+1)+DIB
      ST(K3+2)=ST(K3+2)+DIB
      GO TO 5
100   RETURN
      END
```

```
      SUBROUTINE ORB(IP,D)
C
C
C     THIS SUBROUTINE DETERMINES THE MEAN ORBITAL ELEMENTS (A, E, I,
C     NODE, OMEGA) OF EACH OF THE PLANETS AT A GIVEN DATE.  THE
C
C     ARGUMENTS ARE DEFINED AS FOLLOWS.
C
C          IP - NUMBER OF PLANET AS GIVEN BY THE FOLLOWING CODE
C                1 - SUN
C                2 - MERCURY
C                3 - VENUS
C                4 - EARTH
C                5 - MARS
C                6 - JUPITER
C                7 - SATURN
C                8 - URANUS
C                9 - NEPTUNE
C               10 - PLUTO
C               11 - EARTH-S MOON
C          D - DATE AT WHICH THE ELEMENTS ARE TO BE EVALUATED
C
C     THE ELEMENTS ARE RETURNED IN THE ARRAY ELMNT AS
C
C          ELMNT(8*IP-15) = I
C          ELMNT(8*IP-14) = NODE
C          ELMNT(8*IP-13) = OMEGA
C          ELMNT(8*IP-12) = E
C          ELMNT(8*IP-10) = A
C     ALSO ELMNT(8*IP- 9) = OMEGA - NODE
C
C
C
      COMMON /BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      FN1(A,B,C,D,X)=A+B*X+C*X*X+D*X*X*X
      FN2(A,B,X)=A+B*X
      T=D/36525.
      PI2=2.*PI
      I=IP-1
      IF(I)180,240,180
180   IF(I-9)181,181,220
181   IT = 2*(I-1)
      IL=4*IT
      ELMNT(IL+6)=FN2(SMJR(IT+1),SMJR(IT+2),T)
      IF(I-4)190,190,200
190   IK=IL
      DO 191 J=1,13,4
      IK = IK + 1
      K=20*(I-1)+J
      ELMNT(IK) = FN1(CN(K),CN(K+1),CN(K+2),CN(K+3),T)
191   CONTINUE
      GO TO 210
200   IK=IL
      DO 201J=1,7,2
      IK=IK+1
      K=10*(I-5)+J
      ELMNT(IK) = FN2(ST(K),ST(K+1),T)
201   CONTINUE
```

```
210   ELMNT(IL+7)=ELMNT(IL+3)-ELMNT(IL+2)
      GO TO 240
220   ELMNT(73)=EMN(13)
      ELMNT(74)=FN1(EMN(1),EMN(2)*36525.,EMN(3),EMN(4),T)
      ITEMP=ELMNT(74)/PI2
      ELMNT(74)=ELMNT(74)-FLOAT(ITEMP)*PI2
      ELMNT(75)=FN1(EMN(5),EMN(6)*36525.,EMN(7),EMN(8),T)
      ITEMP=ELMNT(75)/PI2
      ELMNT(75)=ELMNT(75)-FLOAT(ITEMP)*PI2
      ELMNT(76)=EMN(14)
      ELMNT(78)=EMN(15)
240   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OT2(XL,XP,     DDRM1,CCVM1,CCPSI,CCA,CCI,DDVM1,TTRM7,
     1TTVM7,CCTA1,CCTA7,TL,TINJ,NTDD,TF,NTTT,C3,HHVM2,DLAQ,RALQ,RJ,HHVQM
     2,PTH,VHP,DPA,RAP,HHE,DDAZ,TB,PHI,THI,RAI,AZI,TC,CCE)
      DIMENSION NTDD(5),NTTT(5),XL(6),XP(6),DDR1(3),TTR7(3)
      WRITE(6,900)
      PI = 3.1415926536
      DO 1 I=1,3
      DDR1(I)=XL(I)
    1 TTR7(I)=XP(I)
      AU = 149.5985
      AUDAY = AU/(24.*3600.)*1000000.
      RAD = 57.2957795
      R = CCTA7/6.28318531
      N = R
      X = N
      CCTA7 = (R-X)*6.28318531
      IF(DLO) 30,31,30
   30 DLA = ATAN(DDR1/DLO)
      IF(DLO)32,31,33
   31 DLA = PI/2.
      IF(DDR1) 32,33,33
   32 DLA = DLA + PI
   33 IF(DLA)34,35,35
   34 DLA = 2. *PI + DLA
   35 IF(TLO) 40,41,40
   40 TLA =ATAN(TTR7/TLO)
      IF(TLO)42,41,43
   41 TLA = PI/2.
      IF(TTR7)42,43,43
   42 TLA = TLA + PI
   43 IF(TLA)44,45,45
   44 TLA = 2. * PI + TLA
   45 AZL = 0.0
      GAL = 0.0
      GAP = 0.0
      CCVM7 = 0.0
      HCA = 0.0
      AZP = 0.0
      RCA = 0.0
      APO = 0.0
      DDRM1 = DDRM1*AU
      CCVM1 = CCVM1*AUDAY
      CCPSI = CCPSI*RAD
      CCA=CCA*AU
      CCI=CCI*RAD
      DDVM1 = DDVM1*AUDAY
      TTRM7 = TTRM7*AU
      TTVM7 = TTVM7*AUDAY
      CCTA1 = CCTA1*RAD
      CCTA7 = CCTA7*RAD
      WRITE(6,956)NTDD,SDD,TF,NTTT,STT
      WRITE(6,957)
      WRITE(6,950) DDRM1,DLA,DLO,CCVM1,GAL,AZL,CCPSI,CCA,CCE,CCI,DDVM1
      WRITE(6,951) TTRM7,TLA,TLO,CCVM7,GAP,AZP,CCTA1,CCTA7,RCA,APO,TTVM7
      WRITE(6,958)
      WRITE(6,953)C3,HHVM2,DLAQ,RALQ,RJ,HHVQM,PTH,VHP,DPA,RAP,HHE
      WRITE(6,954)
      SEC = TL*3600.
      ITL = TL
      X1 = ITL
```

```
      SECH = X1*3600.
      MTL = (SEC - SECH) /60.
      SECM = MTL*60
      SJTL = SEC - SECH - SECM
      SEC = TINJ * 3600.
      IHOUR = TINJ
      X1 = IHOUR
      SECH = X1*3600.
      MIN = (SEC-SECH)/60.
      SECM = MIN*60
      SECI = SEC-SECH-SECM
      WRITE(6,955)DDAZ,ITL,MTL,SJTL,TB,PHI,THI,RAI,AZI,IHOUR,MIN,SECI,TC
  900 FORMAT(//10X,25HPOINT-TO-POINT CONDITIONS)
  950 FORMAT(6X,2HRL,F8.2,1X,3HLAL,F6.2,1X,3HLOL,F7.2,1X,2HVL,F8.3,1X,3
     1HGAL,F7.2,1X,3HAZL,F7.2,1X,3HHCA,F7.2,1X,3HSMA,F7.2,1X,3HECC,F7.5,
     21X,3HINC,F7.4,1X,2HV1,F8.3)
  951 FORMAT(6X,2HRP,F8.2,1X,3HLAP,F6.2,1X,3HLOP,F7.2,1X,2HVP,F8.3,1X,3
     1HGAP,F7.2,1X,3HAZP,F7.2,1X,3HTAL,F7.2,1X,3HTAP,F7.2,1X,3HRCA,F7.2,
     21X,3HAPO,F7.2,1X,2HV2,F8.3)
  952 FORMAT( 6X,2HRC,F8.3,1X,2HGL,F7.2,1X,2HGP,F8.2,1X,3HZAL,F7.2,1X,3H
     1ZAP,F7.2,1X,3HETS,F7.2,1X,3HZAE,F7.2,1X,3HETE,F7.2,1X,3HZAC,F7.2,1
     2X,3HETC,F7.2,1X,3HCLP,F7.2)
  953 FORMAT(6X,2HC3,F8.3,1X,3HVHL,F8.3,1X,3HDLA,F7.2,1X,3HRAL,F7.2,1X,
     13HRAD,F7.1,1X,3HVEL,F7.3,1X,3HPTH,F6.2,1X,3HVHP,F7.3,1X,3HDPA,F6.2
     2,1X,3HRAP,F7.2,1X,3HECC,F7.4)
  955 FORMAT(7X,F7.2,4X,I2,1X,I2,1X,F2.0,2X,F8.2,3X,F6.2,3X,F7.2,4X,F7.2
     1,5X,F7.2,3X,I2,1X,I2,1X,F2.0,4X,F7.1,3X,F8.2,3X,F9.2)
  954 FORMAT(6X,118HLNCH AZMTH  LNCH TIME  L-I TIME  INJ LAT  INJ LONG
     1INJ RT ASC  INJ AZMTH  INJ TIME  PO CST TIM  INJ 2 LAT  INJ 2 LONG
     2)
  956 FORMAT( 6X,11HLAUNCH DATE,I6,4I3,F8.3,15X,11HFLIGHT TIME,F9.2,8X,
     $ 12HARRIVAL DATE,I6,4I3,F8.3)
  957 FORMAT( 7X,18HHELIOCENTRIC CONIC,33X,8HDISTANCE,2X,F8.3)
  958 FORMAT(7X,20HPLANETOCENTRIC CONIC)
      RETURN
      END
```

```
      SUBROUTINE OUT1(ITARG,JOPT,NITS,NB,IDAT1,S1,IDAT2,S2,IDAT3,S3,BDT,
     2 BDR,DINC,RCA,TOL1,TOL2,TOL3,ACC,RS,INPR,DELTP,NBOD,LVOPT,AC,MIDI)
      DIMENSION NB(11),IDAT1(5),IDAT2(5),IDAT3(5),RS(6),AC(5)
    7 WRITE(6,617)JOPT
      GO TO (9,10),JOPT
    9 WRITE(6,629)
      GO TO 11
   10 WRITE(6,630)
   11 WRITE(6,610)NB(2)
      WRITE(6,611)IDAT1,S1
      IF(ITARG-2)12,12,8
    8 WRITE(6,620)
      RSM=SQRT(RS(1)*RS(1)+RS(2)*RS(2)+RS(3)*RS(3))
      VSM=SQRT(RS(4)*RS(4)+RS(5)*RS(5)+RS(6)*RS(6))
      WRITE(6,621)(RS(I),I=1,3),RSM
      WRITE(6,622)(RS(I),I=4,6),VSM
   12 CONTINUE
    2 WRITE(6,609)ITARG
      GO TO (21,22,23,24,25,26),ITARG
   21 WRITE(6,623)
      GO TO 27
   22 WRITE(6,624)
      GO TO 27
   23 WRITE(6,625)
      GO TO 27
   24 WRITE(6,626)
      GO TO 27
   25 WRITE(6,627)
      GO TO 27
   26 WRITE(6,628)
   27 CONTINUE
      WRITE(6,610)NB(3)
      WRITE(6,611)IDAT3,S3
      GO TO (12,6,6,6,3,3),ITARG
    3 WRITE(6,612)DINC,RCA,IDAT3,S3
      IF(ITARG-5)5,5,4
    4 WRITE(6,613)TOL1,TOL2,TOL3
      TOL1=25.
      TOL2=25.
      TOL3=.005
    5 WRITE(6,614)
    6 WRITE(6,615)BDT,BDR,IDAT2,S2
      WRITE(6,616)TOL1,TOL2,TOL3
      WRITE(6,631)LVOPT
      IF(LVOPT-1)30,30,31
   30 WRITE(6,632)
      WRITE(6,633)ACC,NITS
      GO TO 40
   31 WRITE(6,634)LVOPT
      I8=8
      WRITE(6,633)AC(1),I8
      LV1=LVOPT-1
      DO 32 I=2,LV1
   32 WRITE(6,635)I,AC(I),MIDI
      WRITE(6,635)LVOPT,ACC,NITS
   40 CONTINUE
      WRITE(6,600)
  600 FORMAT(1H1)
  603 FORMAT(//  10X,18HPROGRAM PARAMETERS)
  604 FORMAT(20X,9HACCURACY=,E10.2)
```

```
605 FORMAT(20X,7HBODIES=,11I3)
606 FORMAT(20X,21HMAX NO OF ITERATIONS=,I3)
607 FORMAT(20X,17HPRINT INCREMENTS=,I6)
608 FORMAT(20X,20HPRINT TIME INTERVAL=,F10.3)
609 FORMAT(//10X,27HTARGET CONDITIONS   (OPTION,I2,1H))
610 FORMAT(20X,7HPLANET=,I3)
611 FORMAT(20X,5HDATE=,I5,I3,I3,I3,I3,F7.3)
612 FORMAT(20X,4HINC=,F10.4,5X,4HRCA=,F10.1,5X,4HTCA=,I5,I3,I3,I3,I3,
   1 F7.3)
613 FORMAT(20X,4HTOL=,F10.4,5X,4HTOL=,F10.1,5X,4HTOL=,F9.5)
614 FORMAT(15X,20HAUXILIARY CONDITIONS)
615 FORMAT(20X,4HB.T=,F10.1,5X,4HB.R=,F10.1,5X,4HTSI=,I5,I3,I3,I3,I3,
   1 F7.3)
616 FORMAT(20X,4HTOL=,F10.1,5X,4HTOL=,F10.1,5X,4HTOL=,F9.5)
617 FORMAT(//10X,30HINJECTION CONDITIONS   (OPTION,I2,1H))
618 FORMAT(20X,39HZERO ITERATE INJECTION CONDITIONS INPUT)
619 FORMAT(20X,72HZERO ITERATE INJECTION CONDITIONS COMPUTED FROM PATC
   1HED CONIC TRAJECTORY)
620 FORMAT(20X,33HHELIOCENTRIC ECLIPTIC COORDINATES)
621 FORMAT(20X,9HPOSITION=,3(5X,E17.10),5X,4HMAG=,E17.10)
622 FORMAT(20X,9HVELOCITY=,3(5X,F13.9, 4X),5X,4HMAG=,F14.10)
623 FORMAT(20X,38HOPTION 1 ... POINT-TO-POINT CONDITIONS)
624 FORMAT(20X,46HOPTION 2 ... TARGETED PATCHED CONIC CONDITIONS)
625 FORMAT(20X,36HOPTION 3 ... B.T,B.R,APPROXIMATE TSI)
626 FORMAT(20X,24HOPTION 4 ... B.T,B.R,TSI)
627 FORMAT(20X,36HOPTION 5 ... APPROXIMATE INC,RCA,TCA)
628 FORMAT(20X,24HOPTION 6 ... INC,RCA,TCA)
629 FORMAT(20X,59HOPTION 1 ... COMPUTE R,V,TINJ FROM PATCHED CONIC TRA
   1JECTORY)
630 FORMAT(20X,27HOPTION 2 ... INPUT R,V,TINJ)
631 FORMAT(//10X,28HTARGETING SCHEDULE   (OPTION,I2,1H))
632 FORMAT(20X,7H1 LEVEL)
633 FORMAT(20X,23HLEVEL 1 ..... ACCURACY=, E10.2,5X,11HITERATIONS=,I3,
   1 5X,12HSTM COMPUTED)
634 FORMAT(20X,I1,7H LEVELS)
635 FORMAT(20X,5HLEVEL,I2,16H ..... ACCURACY=,E10.2,5X,11HITERATIONS=,
   1 I3,5X,11HSTM ASSUMED)
    RETURN
    END
```

```
SUBROUTINE PARTL(R,V,B,BDT,BDR,PBT,PBR)
DIMENSION R(3),V(3),PBT(6),PBR(6)
U2=V(1)*V(1)+V(2)*V(2)
U=SQRT(U2)
V2=U2+V(3)*V(3)
S=SQRT(V2)
H3=R(1)*V(2)-V(1)*R(2)
RU=R(1)*V(1)+R(2)*V(2)
U2PV2=U2+V2
UV=U*S
UV3=UV*UV*UV
BDT=H3/U
BDR=(RU*V(3)-U2*R(3))/UV
B=SQRT(BDT*BDT+BDR*BDR)
PBT(1)=V(2)/U
PBT(2)=-V(1)/U
PBT(3)=0.
PBT(4)=-V(2)*RU/(U*U*U)
PBT(5)=V(1)*RU/(U*U*U)
PBT(6)=0.
PBR(1)=V(1)*V(3)/UV
PBR(2)=V(2)*V(3)/UV
PBR(3)=-U/S
PBR(4)=V(3)*(U2*(V2*R(1)-V(1)*R(3)*V(3))-V(1)*U2PV2*RU)/UV3
PBR(5)=V(3)*(U2*(V2*R(2)-V(2)*R(3)*V(3))-V(2)*U2PV2*RU)/UV3
PBR(6)=U*(R(1)*V(1)+R(2)*V(2)+R(3)*V(3))/(S*S*S)
RETURN
END
```

```
      SUBROUTINE PCTM(RI)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RS(3),VS(3),DUM(6,6)
      D=TRTM1+DATEJ
      DO 10 I=1,NBOD
      IP=NB(I)
      NO(1)=IP
      CALL ORB(IP,D)
      CALL EPHEM(1,D,1)
      DO 50 K=1,3
      RS(K)=RI(K)-XP(K)*ALNGTH
   50 VS(K)=RI(K+3)-XP(K+3)*ALNGTH/TM
      RM=SQRT(RS(1)*RS(1)+RS(2)*RS(2)+RS(3)*RS(3))
      IF(SPHERE(IP)*ALNGTH*1.1.GE.RM) GO TO 20
   10 CONTINUE
      IP=1
      DO 60 I=1,3
      RS(I)=RI(I)
   60 VS(I)=RI(I+3)
   20 GMS=PMASS(IP)*ALNGTH*ALNGTH*ALNGTH/(TM*TM)
      DELT=DELTM*TM
      CALL CONC2(RS,VS,DELT,GMS,DUM)
      DO 40 I=1,6
      DO 40 J=1,6
   40 PSI(I,J)=DUM(I,J)
      RETURN
      END
```

```
      SUBROUTINE PECEQ(NP,D,ECEQ)
      DIMENSION ECEQ(3,3),ECOP(3,3),OPEQ(3,3)
      DGTR=.0174532924
      DD=D/10000.
      T=D/36525.
      XLQ=0.
      XIQ=0.
      GO TO (50,2,3,4,5,6,7,8,9,10),NP
    2 XI=0.1222233228+3.24776685E-05*T-3.199770295E-07*T*T
      XL=0.8228518595+2.068578774E-02*T+3.034933644E-06*T*T
      GO TO 20
    3 XI=.0592300268+1.755510339E-05*T-1.696847884E-08*T*T
      XL=1.32260435+1.570534527E-02*T+7.155849933E-06*T*T
      GO TO 20
    4 XI=0.
      XL=0.
      XIQ=23.445*DGTR
      XLQ=0.
      GO TO 20
    5 XI=.0322944089-1.178097245E-05*T+2.201054112E-07*T*T
      XL=.8514840375+1.345634309E-02*T-2.424068406E-08*T*T
     $ -9.308422677E-08*T*T*T
      XIQ=23.99*DGTR
      XLQ=140.881*DGTR
      GO TO 20
    6 XI=0.0228410270-9.696273622E-05*T
      XL=1.7355180770+1.764479392E-02*T
      GO TO 20
    7 XI=0.0435037861-7.757018898E-08*T
      XL=1.9684445802+1.523977870E-02*T
      GO TO 20
    8 XI=0.0134865470+9.696273622E-06*T
      XL=1.2826407707-1.599885148E-04*T
      GO TO 20
    9 XI=0.0310537707-1.599885148E-04*T
      XL=2.2810642235+1.923032859E-02*T
      GO TO 20
   10 XI=0.2996712872
      XL=1.917865870
   20 CALL EULMX(XL,3,XI,1,0.,0,ECOP)
      CALL EULMX(XLQ,-3,XIQ,-1,0.,0,OPEQ)
      DO 60 I = 1,3
      DO 60 J = 1,3
      ECEQ(I,J) = 0.0
      DO 60 K = 1,3
60    ECEQ(I,J) = ECEQ(I,J) + OPEQ(I,K)*ECOP(K,J)
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLANE(XL,XP,HCA,HCI,HCW,HCN,NTYS)
      DIMENSION XL(6),XP(6),HCN(3)
      PI=3.1415926536
      RL=SQRT(XL(1)*XL(1)+XL(2)*XL(2)+XL(3)*XL(3))
      RP=SQRT(XP(1)*XP(1)+XP(2)*XP(2)+XP(3)*XP(3))
      HCN(1)=XL(2)*XP(3)-XL(3)*XP(2)
      HCN(2)=XL(3)*XP(1)-XL(1)*XP(3)
      HCN(3)=XL(1)*XP(2)-XL(2)*XP(1)
      HNM=SQRT(HCN(1)*HCN(1)+HCN(2)*HCN(2)+HCN(3)*HCN(3))
      SGN=1.
      IF(HCN(3))4,4,5
    4 SGN=-1.
    5 HCN(1)=SGN*HCN(1)/HNM
      HCN(2)=SGN*HCN(2)/HNM
      HCN(3)=SGN*HCN(3)/HNM
      RLP=XL(1)*XP(1)+XL(2)*XP(2)+XL(3)*XP(3)
      CHCA=RLP/(RL*RP)
      SHCA=SGN*SQRT(1.-CHCA*CHCA)
      IF(ABS(SHCA)-.0001)6,6,7
    6 LOC=8
      WRITE(3,17)
      RETURN
    7 CI=HCN(3)
      SI=SQRT(1.-CI*CI)
      CW=XL(1)/RL
      SW=XL(2)/RL
      S=SHCA
      C=CHCA
      J=0
    8 J=J+1
      IF(C)9,10,9
    9 ANG=ATAN(S/C)
      IF(C)11,10,12
   10 ANG=PI/2.
      IF(S)11,12,12
   11 ANG=ANG+PI
   12 IF(ANG)13,14,14
   13 ANG=ANG+2.*PI
   14 GO TO (15,16),J
   15 HCA=ANG
      S=SW
      C=CW
      GO TO 8
   16 HCW=ANG
      NTYS=1
      IF(HCA-PI)19,19,18
   18 NTYS=2
   19 IF(XP(3)*(PI-HCA))20,20,21
   20 HCW=HCW+PI
   21 HCI=ATAN(SI/CI)
   17 FORMAT(1X,22HHCA IS LESS THAN .0001)
      RETURN
      END
```

```
      SUBROUTINE PLND(RI,RF)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION THTWG(6,3),RPER(6)
      DIMENSION RI(6),RF(6)
      K=2*(NTP-2)
      IPR=IPRINT
      IPRINT=1
      SAVE1=SMJR(K+1)
      SAVE2=SMJR(K+2)
      SMJR(K+1)=SAVE1+DELAXS/ALNGTH
      SMJR(K+2)=SAVE2+DELAXS/ALNGTH
      CALL NTM(RI,RPER,NTMC,0)
      DELXA=RPER(1)-RF(1)
      DELYA=RPER(2)-RF(2)
      DELZA=RPER(3)-RF(3)
      DO 10 I=1,6
10    THTWG(I,1)=(RPER(I)-RF(I))/DELAXS
      SMJR(K+1)=SAVE1
      SMJR(K+2)=SAVE2
      IF(NTP.GT.5) GO TO 40
      K=20*NTP-28
      SAVE1=CN(K+1)
      SAVE2=CN(K+2)
      SAVE3=CN(K+3)
      SAVE4=CN(K+4)
      CN(K+1)=SAVE1+DELECC
      CN(K+2)=SAVE2+DELECC
      CN(K+3)=SAVE3+DELECC
      CN(K+4)=SAVE4+DELECC
      CALL NTM(RI,RPER,NTMC,0)
      DELXE=RPER(1)-RF(1)
      DELYE=RPER(2)-RF(2)
      DELZE=RPER(3)-RF(3)
      DO 20 I=1,6
20    THTWG(I,2)=(RPER(I)-RF(I))/DELECC
      CN(K+1)=SAVE1
      CN(K+2)=SAVE2
      CN(K+3)=SAVE3
      CN(K+4)=SAVE4
      K=20*(NTP-2)
      SAVE1=CN(K+1)
      SAVE2=CN(K+2)
      SAVE3=CN(K+3)
      SAVE4=CN(K+4)
      CN(K+1)=SAVE1+DELICL
      CN(K+2)=SAVE2+DELICL
      CN(K+3)=SAVE3+DELICL
      CN(K+4)=SAVE4+DELICL
```

```
      CALL NTM(RI,RPER,NTMC,0)
      DELXI=RPER(1)-RF(1)
      DELYI=RPER(2)-RF(2)
      DELZI=RPER(3)-RF(3)
      DO 30 I=1,6
30    THTWG(I,3)=(RPER(I)-RF(I))/DELICL
      CN(K+1)=SAVE1
      CN(K+2)=SAVE2
      CN(K+3)=SAVE3
      CN(K+4)=SAVE4
      GO TO 70
40    K=10*NTP-14
      SAVE1=ST(K+1)
      SAVE2=ST(K+2)
      ST(K+1)=SAVE1+DELECC
      ST(K+2)=SAVE2+DELECC
      CALL NTM(RI,RPER,NTMC,0)
      DELXE=RPER(1)-RF(1)
      DELYE=RPER(2)-RF(2)
      DELZE=RPER(3)-RF(3)
      DO 50 I=1,6
50    THTWG(I,2)=(RPER(I)-RF(I))/DELECC
      ST(K+1)=SAVE1
      ST(K+2)=SAVE2
      K=10*(NTP-2)
      SAVE1=ST(K+1)
      SAVE2=ST(K+2)
      ST(K+1)=SAVE1+DELICL
      ST(K+2)=SAVE2+DELICL
      CALL NTM(RI,RPER,NTMC,0)
      DELXI=RPER(1)-RF(1)
      DELYI=RPER(2)-RF(2)
      DELZI=RPER(3)-RF(3)
      DO 60 I=1,6
60    THTWG(I+3)=(RPER(I)-RF(I))/DELICL
      ST(K+1)=SAVE1
      ST(K+2)=SAVE2
70    IF(IAUG-9) 80,90,100
80    K=6
      GO TO 110
90    K=8
      GO TO 110
100   K=12
110   DO 111 I=1,6
      DO 111 J=1,3
      KJ=K+J
111   PSI(I,KJ)=THTWG(I,J)
      IPRINT=IPR
150   RETURN
      END
```

```
      SUBROUTINE POSVL(A,E,XI,WC,W,AM,WP,RP,R,VP,V,GMS)
      DIMENSION RP(3),VP(3),WP(3),WRP(3)
      P=A*(1.-E*E)
      SNI=SIN(XI)
      CSI=COS(XI)
      SNWC=SIN(WC)
      CSWC=COS(WC)
   12 IF(AM)10,11,11
   10 AM=AM+6.2831853072
      GO TO 12
   11 R=AM/6.28318531
      N=R
      X=N
      AM=(R-X)*6.28318531
      EA=AM+E*SIN(AM)+.5*E**2*SIN(2.*AM)
      DO 20 I=1,10
      SEA=SIN(EA)
      CEA=COS(EA)
      AM1=EA-E*SEA
      DLE=(AM-AM1)/(1.-E*CEA)
      IF(ABS(DLE)-.5E-7)2,20,20
   20 EA=EA+DLE
      WRITE(3,4)DLE
    4 FORMAT(13H0NO CONV DLE=E15.8)
    2 SGN=SEA/ABS(SEA)
      R=A*(1.-E*CEA)
      CTA=(P-R)/(E*R)
      STA=SQRT(1.-CTA**2)*SGN
      PI = 3.1415926536
      IF(CTA)30,31,30
   30 TA = ATAN(STA/CTA)
      IF(CTA)32,31,33
   31 TA = PI/2.
      IF(STA)32,33,33
   32 TA = TA + PI
   33 IF(TA)34,35,35
   34 TA = 2.*PI + TA
   35 CSWV=COS(W+TA)
      SNWV=SIN(W+TA)
      RP(1)=R*CSWV*CSWC-R*SNWV*SNWC*CSI
      RP(2)=R*CSWV*SNWC+R*SNWV*CSWC*CSI
      RP(3)=R*SNWV*SNI
      V=SQRT(2.*GMS/R-GMS/A)
      CSG=SQRT(GMS*P)/(R*V)
      SNG=SGN*SQRT(1.-CSG**2)
      WRP(1)=WP(2)*RP(3)-WP(3)*RP(2)
      WRP(2)=WP(3)*RP(1)-WP(1)*RP(3)
      WRP(3)=WP(1)*RP(2)-WP(2)*RP(1)
      VP(1)=V/R*(WRP(1)*CSG+RP(1)*SNG)
      VP(2)=V/R*(WRP(2)*CSG+RP(2)*SNG)
      VP(3)=V/R*(WRP(3)*CSG+RP(3)*SNG)
      RETURN
      END
```

```
      SUBROUTINE PRED(RI,TEVN)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR THE LOGIC USED IN A PREDICTION
C     EVENT.
C
C     PRED USES THE FOLLOWING SUBROUTINES
C        NTM
C        PSIM
C        DYNO
C        NAVM
C        JACOBI
C        HYELS
C
C
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),P1(17,17),PEIG(3,3),EGVL(3),EGVCT(3,3)
      DIMENSION VEIG(9),RHO(17,17),DUM(2,2),DUM2(2,2)
      DIMENSION DUM3(2)
C     CALCULATE P(TEVN,TRTM1)
      MAX=60
      NPE=NPE+1
      TPT=TPT2(NPE)
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
301   DO 4 I=1,6
4     XF(I)=RF(I)
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      WRITE(6,3007) TEVN
      WRITE(6,3001) (CMPNM(IAUG,I),XF(I),I=1,NDIM)
      LINES=NDIM+10
      CALL PSIM(RI,RF,ISTMC)
      WRITE (6,3002) TEVN,TRTM1
      LINES=LINES+5
      DO 1 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 7
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
7     IF (NDIM.EQ.6) GO TO 8
      WRITE (6,3013) I
```

```
      LINES=LINES+1
8     WRITE (6,3014) (PSI(I,J),J=1,NDIM)
1     LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(0)
      IF (LINES.LE.MAX-8 ) GO TO 12
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
12    WRITE (6,3003)
      WRITE (6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF(LINES.LT.MAX-9) GO TO 13
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
13    WRITE (6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 2 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 9
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
9     IF (NDIM.EQ.6) GO TO 23
      WRITE (6,3013) I
      LINES=LINES+1
23    WRITE (6,3014) (P(I,J),J=1,NDIM)
2     LINES=LINES+(NDIM-1)/6+1
      DO 5 I=1,6
5     RI(I)=RF(I)
      DO 10 I=1,NDIM
      DO 10 J=1,NDIM
10    P1(I,J)=P(I,J)
      TRTM1=TEVN
      DELTM=TPT-TEVN
      GO TO (200,200,201,200,201,200,201,200,201,201,201), IAUG
200   IF(ISTMC.NE.3) GO TO 15
201   IPR=IPRINT
      IPRINT=1
C     CALCULATE P(TPT,TEVN)
      CALL NTM(RI,RF,NTMC,0)
      IPRINT=IPR
15    CALL PSIM(RI,RF,ISTMC)
      IF(LINES.LT.MAX-9) GO TO 16
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
16    WRITE (6,3002) TPT,TEVN
      LINES=LINES+5
      DO 3 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 24
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
24    IF (NDIM.EQ.6) GO TO 25
      WRITE (6,3013) I
      LINES=LINES+1
25    WRITE (6,3014) (PSI(I,J),J=1,NDIM)
3     LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(0)
```

```
      IF (LINES.LE.MAX-8) GO TO 17
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
17    WRITE (6,3003)
      WRITE (6,3014)(Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF(LINES.LT.MAX-9) GO TO 18
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
18    WRITE (6,3006) TPT,TEVN
      LINES=LINES+5
      DO 6 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 26
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
26    IF (NDIM.EQ.6) GO TO 27
      WRITE(6,3013) I
      LINES=LINES+1
27    WRITE (6,3014) (P(I,J),J=1,NDIM)
6     LINES=LINES+(NDIM-1)/6+1
50    ICODE=0
      DO 60 I=1,3
      DO 60 J=1,3
60    PEIG(I,J)=P(I,J)
      K=0
      DO 98 J=1,3
      DO 98 I=1,3
      K=K+1
98    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF (LINES.LE.MAX-16) GO TO 21
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
21    WRITE (6,1000) (I,EGVL(I),I=1,3)
      WRITE (6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
65    IF(IHYP1-2) 70,80,70
70    IF(LINES.LT.MAX-16) GO TO 71
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
71    CALL HYELS(1,PEIG,3)
      LINES=LINES+16
80    IF(IHYP1-1) 100,100,90
90    IF(LINES.LT.MAX-16) GO TO 91
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
91    CALL HYELS(3,PEIG,3)
      LINES=LINES+16
100   IF(ICODE)105,105,130
105   IF(IEIG) 130,130,110
110   DO 120 I=1,3
      DO 120 J=1,3
120   PEIG(I,J)=P(I+3,J+3)
```

```
      K=0
      DO 99 J=4,6
      DO 99 I=4,6
      K=K+1
99    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
      IF (LINES.LE.MAX-16) GO TO 22
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
22    WRITE(6,2000) (I,EGVL(I),I=1,3)
      WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      ICODE=1
      GO TO 65
130   DO 500 I=1,NDIM
      DO 500 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
500   RHO(J,I)=RHO(I,J)
      IF(LINES.LT.MAX-9) GOTO 501
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
501   WRITE(6,3020) TEVN
      LINES=LINES+5
      DO 504 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 502
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
502   IF(NDIM.EQ.6) GO TO 503
      WRITE(6,3013) I
      LINES=LINES+1
503   WRITE(6,3014) (RHO(I,J),J=1,NDIM)
504   LINES=LINES+(NDIM-1)/6+1
      IF(NGE.EQ.0) GO TO 139
      IF(ABS(TSOI1    -TPT).GT.1.) GO TO 139
      DO 135 I=1,2
      DO 135 J=1,2
      DUM(I,J)=0.
      DO 135 K=1,6
      DO 135 L=1,6
135   DUM(I,J)=DUM(I,J)+EM(I,K)*P(K,L)*EM(J,L)
      IF(LINES.LT.MAX-24) GO TO 136
      IPGN=IPGN+1
      WRITE (6,3000) (MDNM(ITR,I),I=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
136   WRITE(6,3015) ((DUM(I,J),J=1,2),I=1,2)
3015  FORMAT(///8X*COVARIANCE OF UNCERTAINTIES IN B DOT T AND B DOT R AT
     $ SPHERE OF INFLUENCE*//2(10X2E25.13/))
      CALL JACOBI(DUM,DUM3,DUM2,2,FOV)
      WRITE(6,3016) (I,DUM3(I),I=1,2),(I,(DUM2(I,J),J=1,2),I=1,2)
3016  FORMAT(///20X*EIGENVALUES OF ABOVE MATRIX*//2(22X,I2,E20.10/)///
     $20X*EIGENVECTORS OF ABOVE MATRIX*//2(22X,I2,2E20.10/))
139   DO 140 I=1,NDIM
      XI(I)=XF(I)
      DO 140 J=1,NDIM
140   P(I,J)=P1(I,J)
1000  FORMAT(///20X*POSITION EIGENVALUES */3(22X,I2, E20.10/))
1001  FORMAT(///20X*POSITION EIGENVECTORS*/3(22X,I2,3E20.10/))
```

```
2000  FORMAT(///20X*VELOCITY EIGENVALUES */3(22X,I2,E20.10/))
2001  FORMAT(///20X*VELOCITY EIGENVECTORS*/3(22X,I2,3E20.10/))
3000  FORMAT(1H1//5X2A10*-- PREDICTION EVENT AT TRAJECTORY TIME*F12.3
     $* DAYS, PREDICTING TO TRAJECTORY TIME*F12.3* DAYS*/90X*PROBLEM. .*
     $I10,5X*PAGE. .*I8///1X,130(1H*)//)
3001  FORMAT(10X,A10,E20.13)
3002  FORMAT(///8X*STATE TRANSITION MATRIX  -- PSI(*F12.3*,*F12.3*)*/)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF PREDICTION EVENT  --  P(
     $*F12.3*,*F12.3*)*/)
3006  FORMAT(///8X*COVARIANCE MATRIX AT PREDICTION TIME  --  P(*F12.3*,*
     $F12.3*)*/)
3007  FORMAT(8X*STATE VECTOR AT TIME *F12.3* DAYS*/)
3013  FORMAT (10X*ROW*I3)
3014  FORMAT(16X,6E17.8)
3020  FORMAT(///8X*CORRELATION COEFFICIENT MATRIX AT PREDICTION TIME-- *
     $F8.3* DAYS*/)
      RETURN
      END
```

```
      SUBROUTINE PRESIM(RI,TEVN,RI1)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON /CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),RI1(6),RF1(6),RI2(6),RF2(6),DUM(17),DM2(2,2)
      DIMENSION PEIG(3,3),EGVL(3),EGVCT(3,3),VEIG(9),RHO(17,17),DM(2,2)
      DIMENSION P1(17,17),DM3(2)
      MAX=60
      NPE=NPE+1
      TPT=TPT2(NPE)
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
      DO 10 I=1,6
10    XF(I)=RF(I)
      IF (NQE.NE.0) GO TO 20
      DO 11 I=1,NDIM
11    XF1(I)=XF(I)
      DO 12 I=1,6
12    RF1(I)=RF(I)
      GO TO 30
20    CALL NTM(RI1,RF1,NTMC,2)
      DO 21 I=1,6
21    XF1(I)=RF1(I)
30    CALL PSIM(RI1,RF1,ISTMC)
      CALL DYNO(0)
      CALL NAVM(1,1)
      DO 50 I=1,NDIM
      DO 50 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
50    RHO(J,I)=RHO(I,J)
      DO 39 I=1,6
39    RI2(I)=XI1(I)+ADEVX(I)
      CALL NTM(RI2,RF2,NTMC,3)
      DO 40 I=1,6
40    Z(I)=RF2(I)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
      WRITE (6,3006)
      LINES=LINES+3
```

```
      WRITE(6,3001) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
      LINES=LINES+NDIM
      WRITE(6,3002) TEVN,TRTM1
      LINES=LINES+5
      DO 33 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 31
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
31    IF (NDIM.EQ.6) GO TO 32
      WRITE(6,3013) I
      LINES=LINES+1
32    WRITE(6,3014) (PSI(I,J),J=1,NDIM)
33    LINES=LINES+(NDIM-1)/6+1
      IF(LINES.LT.MAX-8) GO TO 34
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
34    WRITE(6,3003)
      WRITE(6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      IF (LINES.LT.MAX-9) GO TO 35
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
35    WRITE(6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 38 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 36
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
36    IF (NDIM.EQ.6) GO TO 37
      WRITE(6,3013) I
      LINES=LINES+1
37    WRITE(6,3014) (P(I,J),J=1,NDIM)
38    LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(1)
      DO 43 I=1,6
43    ADEVX(I)=Z(I)+W(I)-XF1(I)
      DO 45 I=1,NDIM
      DUM(I)=0.
      DO 45 J=1,NDIM
45    DUM(I)=DUM(I)+PSI(I,J)*EDEVX(J)
      DO 46 I=1,NDIM
46    EDEVX(I)=DUM(I)
      ICODE1=1
48    ICODE=0
      K=0
      DO 60 J=1,3
      DO 60 I=1,3
      K=K+1
      PEIG(I,J)=P(I,J)
60    VEIG(K)=P(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOP)
      IF (LINES.LT.MAX-16) GO TO 62
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
62    WRITE(6,1000) (I,EGVL(I),I=1,3)
```

```
      WRITE(6,1001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
65    IF (IHYP1-2) 70,80,70
70    IF (LINES.LT.MAX-15) GO TO 71
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
71    CALL HYELS(1,PEIG,3)
      LINES=LINES+14
80    IF(IHYP1-1) 100,100,90
90    IF(LINES.LT.MAX-15) GO TO 91
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
91    CALL HYELS(3,PEIG,3)
      LINES=LINES+14
100   IF(ICODE) 105,105,130
105   IF(IEIG) 130,130,110
110   K=0
      DO 120 J=1,3
      DO 120 I=1,3
      K=K+1
      PEIG(I,J)=P(I+3,J+3)
120   VEIG(K)=PEIG(I,J)
      CALL JACOBI(VEIG,EGVL,EGVCT,3,FOV)
      IF (LINES.LT.MAX-16) GO TO 121
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
121   WRITE(6,2000) (I,EGVL(I),I=1,3)
      WRITE(6,2001) (I,(EGVCT(I,J),J=1,3),I=1,3)
      LINES=LINES+16
      ICODE=1
      GO TO 65
130   GO TO (131,230),ICODE1
131   IF (LINES.LT.MAX-9) GO TO 51
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
51    WRITE(6,3010) TEVN
      LINES=LINES+5
      DO 54 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 52
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
52    IF(NDIM.EQ.6) GO TO 53
      WRITE(6,3013) I
      LINES=LINES+1
53    WRITE(6,3014) (RHO(I,J),J=1,NDIM)
54    LINES=LINES+(NDIM-1)/6+1
      IF (LINES.LT.MAX-NDIM-5) GO TO 42
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
42    WRITE(6,3007) (W(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      IF (LINES.LT.MAX-NDIM-7) GO TO 47
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
```

```
      LINES=9
47    WRITE(6,3009) (EDEVX(I),ADEVX(I),I=1,NDIM)
      LINES=LINES+NDIM+7
      DO 132 I=1,6
      RI(I)=RF(I)
132   RI1(I)=RF1(I)
      DO 150 I=1,NDIM
      DO 150 J=1,NDIM
150   P1(I,J)=P(I,J)
      TRTM1=TEVN
      DELTM=TPT-TEVN
      GO TO (160,160,161,160,161,160,161,160,161,161,161),IAUG
160   IF(ISTMC.NE.3) GO TO 170
161   IPR=IPRINT
      IPRINT=1
      CALL NTM(RI1,RF1,NTMC,0)
      IPRINT=IPR
170   CALL PSIM(RI1,RF1,ISTMC)
      IF(LINES.LT.MAX-9) GO TO 171
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
171   WRITE(6,3002) TPT,TEVN
      LINES=LINES+5
      DO 182  I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 180
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
180   IF(NDIM.EQ.6) GO TO 181
      WRITE(6,3013) I
      LINES=LINES+1
181   WRITE(6,3014) (PSI(I,J),J=1,NDIM)
182   LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(0)
      IF(LINES.LT.MAX-8) GO TO 190
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
190   WRITE(6,3003)
      WRITE(6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      CALL NAVM(1,1)
      IF(LINES.LT.MAX-9) GO TO 200
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
200   WRITE(6,3012) TPT,TEVN
      LINES=LINES+5
      DO 203 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 201
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
201   IF (NDIM.EQ.6) GO TO 202
      WRITE(6,3013) I
      LINES=LINES+1
202   WRITE(6,3014) (P(I,J),J=1,NDIM)
203   LINES=LINES+(NDIM-1)/6+1
      DO 210 I=1,NDIM
```

```
      DO 210 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
210   RHO(J,I)=RHO(I,J)
      ICODE1=2
      GO TO 48
230   IF (LINES.LT.MAX-9) GO TO 220
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
220   WRITE(6,3010) TPT
      DO 223 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 221
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
221   IF (NDIM.EQ.6) GO TO 222
      WRITE (6,3013) I
      LINES=LINES+1
222   WRITE(6,3014) (RHO(I,J),J=1,NDIM)
223   LINES=LINES+(NDIM-1)/6+1
      LINES=9
      IF (LINES.LT.MAX-24) GO TO 241
      IF (NGE.EQ.0) GO TO 250
      IF (ABS(TSOI1    -TPT).GT.1.) GO TO 250
      DO 240 I=1,2
      DO 240 J=1,2
      DM(I,J)=0.
      DO 240 K=1,6
      DO 240 L=1,6
240   DM(I,J)=DM(I,J)+EM(I,K)*P(K,L)*EM(J,L)
      IF(LINES.LT.MAX-24) GO TO 241
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,TPT,IPROB,IPGN
      LINES=9
241   WRITE(6,3015) ((DM(I,J),J=1,2),I=1,2)
      CALL JACOBI(DM,DM3   ,DM2,2,FOV)
      WRITE(6,3016) (I, DM3(I),I=1,2),(I,(DM2(I,J),J=1,2),I=1,2)
250   DO 251 I=1,NDIM
      XI(I)=XF(I)
      XI1(I)=XF1(I)
      DO 251 J=1,NDIM
251   P(I,J)=P1(I,J)
      RETURN
1000  FORMAT(///20X*POSITION EIGENVALUES*/3(22X,I2,E20.10/))
1001  FORMAT(///20X*POSITION EIGENVECTORS*/3(22X,I2,3E20.10/))
2000  FORMAT(///20X*VELOCITY EIGENVALUES*/3(22X,I2,E20.10/))
2001  FORMAT(///20X*VELOCITY EIGENVECTORS*/3(22X,I2,3E20.10/))
3000  FORMAT(1H1//8X2A10*--  PREDICTION EVENT AT TRAJECTORY TIME *F12.3,
     $* DAYS, PREDICTING TO TRAJECTORY TIME *F8.3* DAYS*//90X*PROBLEM. .
     $*I10,5X,*PAGE. .*I8///1X,130(1H*)/)
3001  FORMAT(8XA10E20.10,5X,E20.10,5X,E20.10)
3002  FORMAT(///8X*STATE TRANSITION MATRIX  --  PSI(*F8.3*,*F8.3*)*/)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF PREDICTION EVENT  --  P(
     $*F8.3*,*F8.3*)*/)
3006  FORMAT(8X*STATE VECTOR*//22X*ORIGINAL NOMINAL*7X*MOST RECENT NOMIN
     $AL*13X*ACTUAL*)
3007  FORMAT(///8X*ACTUAL DYNAMIC NOISE*//(8XE20.10))
3009  FORMAT(///8X*DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NO
     $MINAL TRAJECTORY*//15X*ESTIMATED*13X*ACTUAL*/(8X2E20.10))
```

```
3010  FORMAT(///8X*CORRELATION COEFFICEINT MATRIX AT TIME OF PREDICTION
     $EVENT *F8.3* DAYS*/)
3013  FORMAT(10X*ROW *I3)
3014  FORMAT(16X6E17.8)
3012  FORMAT(///8X*COVARIANCE MATRIX AT PREDICTION TIME  --  P(*F8.3*,*
     $F8.3*)*/)
3015  FORMAT(///8X*COVARIANCE OF UNCERTAINTIES IN B DOT T AND B DOT R AT
     $ SPHERE OF INFLUENCE*// 2(10X2E25.13/))
3016  FORMAT(///20X*EIGENVALUES OF ABOVE MATRIX*//2(22XI2,E20.10/)///20X
     $*EIGENVECTORS OF ABOVE MATRIX*//2(22X,I2,2E20.10/))
      END
```

```
      SUBROUTINE PRINT
C
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR THE PRINTOUT THROUGHOUT THE
C     TRAJECTORY.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DIMENSION IDATE(6),TMP(11,4)
      CALL NEWPGE
      N2=NBODY+2
      DO 1 I=1,N2,2
      DO 1 J=1,3
      F(I,J)=F(I,J)*V(1,6)
1     F(I+1,J)=F(I+1,J)*V(4,7)
      DO 10 J=1,4
      V(15,J)=V(16,J)*V(5,5)
      DO 10 I=1,11,2
10    V(I,J)=V(I+1,J)
      DO 11 J=2,4
      DO 11 I=1,9,4
      V(I,J)=V(I,J)*V(1,6)
11    V(I+2,J)=V(I+2,J)*V(4,7)
      V(5,1)=V(6,1)*V(6,5)
      V(7,1)=V(8,1)*V(6,6)
      V(9,1)=V(9,1)*V(1,6)
      RV=SQRT(V(5,2)*V(5,2)+V(5,3)*V(5,3)+V(5,4)*V(5,4))
      VV=SQRT(V(7,2)*V(7,2)+V(7,3)*V(7,3)+V(7,4)*V(7,4))
      RS=SQRT(V(1,2)*V(1,2)+V(1,3)*V(1,3)+V(1,4)*V(1,4))
      VS=SQRT(V(3,2)*V(3,2)+V(3,3)*V(3,3)+V(3,4)*V(3,4))
      CALL SPACE(9)
      WRITE(6,1000) V(2,1),INCMNT,(V(1,I),I=2,4),RS,(V(3,I),I=2,4),VS
1000  FORMAT(////* TRAJECTORY TIME = *E20.11,10X*TOTAL TIME INCREMENTS =
     $*,I8//   1X*SPACECRAFT INERTIAL TRAJECTORY* /
     $           5X*POSITION . . . . . . . . . . . . . .*4E20.11/
     $           5X*VELOCITY . . . . . . . . . . . . . .*4E20.11)
      IF (IPRT(2).EQ.0) GO TO 30
      CALL TIME (V(4,1),IYR,MO,IDAY,IHR,MIN,SEC,1)
      D=V(4,1)+2415020.
      MO = MONTH(MO)
      CALL SPACE(3*NBODYI+9)
      WRITE(6,1001)MO,IDAY,IHR,MIN,SEC,IYR,D
1001  FORMAT(//1X130(1H*)///* CALENDAR DATE =*A10,I3*,*I3* HR,*I3* MIN,*
     $F7.3* SEC,*I5/ * JULIAN DATE    = *F17.8// * EPHEMERIS DATA*)
      K=0
      DO 20 I=1,NBODY,4
      K=K+1
      IP=NO(K)
      RP=SQRT(F(I   ,1)*F(I   ,1)+F(I   ,2)*F(I   ,2)+F(I   ,3)*F(I   ,3))
      VP=SQRT(F(I+1,1)*F(I+1,1)+F(I+1,2)*F(I+1,2)+F(I+1,3)*F(I+1,3))
      WRITE(6,1002) PLANET(IP),(F(I ,J),J=1,3),RP,PLANET(IP),(F(I +1,J),
     $J=1,3),VP
```

```
1002  FORMAT (5X*POSITION OF *A10* . . . . . .*4E20.11/5X*VELOCITY OF *
     $A10* . . . . . .*4E20.11/)
20    CONTINUE
30    IF(IPRT(3).EQ.0) GO TO 50
      CALL SPACE(3*NBODYI+6)
      WRITE(6,1003)
1003  FORMAT(//1X,130(1H*)///* SPACECRAFT RELATIVE TRAJECTORIES*)
      K=0
      DO 40 I=1,NBODY,4
      K=K+1
      IP=NO(K)
      F(I+2,4)=F(I+2,4)*V(1,6)
      VSP=SQRT(F(I+3,1)*F(I+3,1)+F(I+3,2)*F(I+3,2)+F(I+3,3)*F(I+3,3))
      WRITE (6,1004) PLANET(IP),(F(I+2,J),J=1,4),PLANET(IP),(F(I  +3,J),
     $J=1,3),VSP
1004  FORMAT (5X*POSITION REL. TO *A10 *  . . .*4E20.11/
     $        5X*VELOCITY REL. TO *A10 *  . . .*4E20.11/)
40    CONTINUE
50    IF(IPRT(4).EQ.0) GO TO 60
      VMR=SQRT(V(11,2)*V(11,2)+V(11,3)*V(11,3)+V(11,4)*V(11,4))
      CALL SPACE(14)
      WRITE (6,1005) (V(5,I),I=2,4),RV,(V(7,I),I=2,4),VV,(V(9,I),I=2,4),
     $V(9,1),(V(11,I),I=2,4),VMR,(V(15,I),I=2,4),V(15,1),(V(14,I),I=2,4)
     $,V(14,1),V(5,1),V(7,1)
1005  FORMAT(//1X,130( 1H*)///* VIRTUAL MASS DATA  *  /
     $             40H     VIRTUAL MASS POSITION                4(  E20.11
     $)/           40H     VIRTUAL MASS VELOCITY                4E20.11/
     $             40H     SPACECRAFT POS. REL. TO V.M.         4E20.11/
     $             40H     SPACECRAFT VEL. REL. TO V.M.         4E20.11/
     $             40H     KEPLER (ANG. MOM.) VECTOR            4E20.11/
     $             40H     ECCENTRICITY VECTOR                  4E20.11/
     $             24H     V.M. MAGN.      =  E20.11/
     $             24H     V.M. MAGN. RATE =  E20.11)
      CALL SPACE(NBODYI+6)
      WRITE(6,1007)
1007  FORMAT(//1X,130(1H*)///* V.M. RELATIVE POSITIONS*)
      DO 52 I=1,NBODYI
      K=NO(I)
      TMP(I,1)=0.
      DO 51 J=2,4
      TMP(I,J)=V(5,J)-F(4*I-3,J-1)
51    TMP(I,1)=TMP(I,1)+TMP(I,J)*TMP(I,J)
      TMP(I,1)=SQRT(TMP(I,1))
52    WRITE(6,1008) PLANET(K),(TMP(I,J),J =2,4),TMP(I,1)
1008  FORMAT (5X*POSITION REL. TO *A10 *  . . .*4E20.11)
60    CALL SPACE(1)
      RETURN
      END
```

```
      SUBROUTINE PRINT1(RF)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR PRINTING A SUMMARY OF THE
C     TRAJECTORY GENERATED IN THE TRAJECTORY MODE.
C
C
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12)
      COMMON /PRT/MONTH(12),PLANET(11)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),VE(6),VT(6)
      F(A,B,C)=SQRT(A*A+B*B+C*C)
      MAX=60
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
1000  FORMAT(1H1///5X*S U M M A R Y   O F   T R A J E C T O R Y   M O D
     $E*/90X*PROBLEM. .*I10,5X*PAGE. .*I8////)
      DO 1 I=1,6
1     RI(I)=XI(I)
      TRTM1=TRTMB
      GO TO (10,20),NTMC
10    WRITE(6,1001) IPROB
1001  FORMAT(8X*PATCHED CONIC TRAJECTORY*10X*PROBLEM*I5///)
      GO TO 100
20    WRITE(6,2001) IPROB
2001  FORMAT(8X*VIRTUAL MASS TRAJECTORY*10X*PROBLEM*I5///)
      WRITE(6,2010) ACC,DELTH
2010  FORMAT(10X*ACCURACY FIGURE* E13.6,* INDICATES TRUE ANOMALY INCREME
     $NT IS* E20.13*  RADIANS*//)
      D1=TRTM1+DATEJ
      TRTM2=TRTM1+DELTM
      D2 = TRTM2+DATEJ
      D3=D1+2415020.
      D4=D2+2415020.
      CALL TIME(D1,LYR,LMO,LDAY,LHR,LMIN,SECL,1)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2002) TRTM1,D3,LMO,LDAY,LHR,LMIN,SECL,LYR,TRTM2,D4,IMO,
     $IDAY,IHR,IMIN,SECI,IYR
2002  FORMAT(10X*INITIAL TRAJECTORY TIME*F12.5* DAYS, JULIAN DATE*F20.10
     $,5X*CALENDAR DATE*4I3,F7.3,*,*I5
     $/     10X*FINAL TRAJECTORY TIME   *F12.5* DAYS, JULIAN DATE*F20.10
     $,5X*CALENDAR DATE*4I3,F7.3,*,*I5)
      RMI=F(RI(1),RI(2),RI(3))
      VMI=F(RI(4),RI(5),RI(6))
      RMF=F(RF(1),RF(2),RF(3))
      VMF=F(RF(4),RF(5),RF(6))
      WRITE(6,2003)
2003  FORMAT(//57X*X-COMP.*13X*Y-COMP.*13X*Z-COMP.*12X*RESULTANT*)
      WRITE(6,2004) (RI(I),I=1,3),RMI,(RI(I),I=4,6),VMI,(RF(I),I=1,3),
     $RMF,(RF(I),I=4,6),VMF
2004  FORMAT(/10X*HELIOCENTRIC ECLIPTIC COORDINATES*//
     $         10X*INITIAL POSITION OF VEHICLE . . . . .*4E20.8/
     $         10X*INITIAL VELOCITY OF VEHICLE . . . . .*4E20.8/
```

```
     $         10X*FINAL POSITION OF VEHICLE . . . . . .*4E20.8/
     $         10X*FINAL VELOCITY OF VEHICLE . . . . . .*4E20.8)
      DO 30 I=1,6
      VT(I)=RF(I)-RTP(I)
30    VE(I)=RF(I)-RE(I)
      RMI=F(VT(1),VT(2),VT(3))
      VMI=F(VT(4),VT(5),VT(6))
      RMF=F(VE(1),VE(2),VE(3))
      VMF=F(VE(4),VE(5),VE(6))
      WRITE(6,2005) (VE(I),I=1,3),RMF,(VE(I),I=4,6),VMF,(VT(I),I=1,3),
     $RMI,(VT(I),I=4,6),VMI
2005  FORMAT(//10X*AT FINAL TIME*//
     $10X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $10X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      RMI=F(RC(1),RC(2),RC(3))
      VMI=F(RC(4),RC(5),RC(6))
      D4=DC+2415020.
      CALL TIME(DC,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2006) IMO,IDAY,IHR,IMIN,SECI,IYR,D4,
     $              (RC(I),I=1,3),RMI,(RC(I),I=4,6),VMI
2006  FORMAT(//10X*AT CLOSEST APPROACH. . . . CALENDAR DATE*4I3,F7.3,*,*
     $I5,*. . .JULIAN DATE *F20.10//
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      IF(ISPH) 40,45,40
40    RMI=F(RSI(1),RSI(2),RSI(3))
      VMI=F(VSI(1),VSI(2),VSI(3))
      D4=DSI+2415020.
      CALL TIME(DSI,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2007) IMO,IDAY,IHR,IMIN,SECI,IYR,D4 ,RSI,RMI,VSI,VMI
2007  FORMAT(//10X*AT SPHERE OF INFLUENCE. . . . CALENDAR DATE*4I3,F7.3,
     $*,*I5,*. . .JULIAN DATE *F20.10//
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      IF(MAX.GT.51) GO TO 41
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
41    WRITE(6,2008) B,BDT,BDR
2008  FORMAT(//10X*B =*E20.8*    B DOT T =*E20.8*    B DOT R =*E20.8)
      IF(MAX.GT.51) GO TO 50
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      GO TO 50
45    WRITE(6,2009)
2009  FORMAT(//10X*VEHICLE DID NOT PIERCE SPHERE OF INFLUENCE OF TARGET
     $PLANET*)
50    WRITE(6,2012) INCMT
2012  FORMAT(//10X*TOTAL NUMBER OF TIME INCREMENTS FOR THIS PROBLEM IS*
     $I6)
      WRITE(6,2011) TIMINT
2011  FORMAT(//10X*TOTAL CP TIME FOR THIS PROBLEM IS* F10.3*  SEC*)
100   RETURN
      END
```

```
      SUBROUTINE PRINT3(MMCODE,NR)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /MEAS/TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON /STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON /TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DATA CMPNM/11*10HRX        ,11*10HRY        ,11*10HRZ        ,11*
     $10HVX        ,11*10HVY        ,11*10HVZ        ,10H          ,10HR
     $ADIUS 1  ,10HMU OF SUN ,10HRANGE BIAS,10HA BIAS    ,3*10HRADIUS 1
     $ ,10HMU OF SUN ,10HRANGE BIAS,10HRADIUS 1  ,10H          ,10HLATIT
     $UDE 1,10HMU-TARG PL,10HR-RAT BIAS,10HECC BIAS  ,3*10HLATITUDE 1,10
     $HMU-TARG PL,10HR-RAT BIAS,10HLATITUDE 1,10H          ,10HLONG 1
     $ ,10H          ,10HSTAR ANG 1,10HINC BIAS  ,3*10HLONG 1    ,10HA B
     $IAS    ,10HSTAR ANG 1,10HLONG 1    ,3*10H          ,10HSTAR ANG 2,
     $10H          ,10HRADIUS 2  ,10HMU OF SUN ,10HRANGE BIAS,10HECC BIA
     $S  ,10HSTAR ANG 2,10HMU OF SUN ,3*10H          ,10HSTAR ANG 3,10H
     $          ,10HLATITUDE 2,10HMU-TARG PL,10HR-RAT BIAS,10HINC BIAS  ,
     $10HSTAR ANG 3,10HMU-TARG PL,3*10H          ,10HAPP DIAM  ,10H
     $     ,10HLONG 2    ,10H          ,10HSTAR ANG 1,10H          ,10HA
     $PP DIAM  ,10HRANGE BIAS,5*10H          ,10HRADIUS 3  ,10H
     $ ,10HSTAR ANG 2,10H          ,10HA BIAS    ,10HR-RAT BIAS,5*10H
     $      ,10HLATITUDE 3,10H          ,10HSTAR ANG 3,10H          ,10
     $HECC BIAS  ,10HSTAR ANG 1,5*10H          ,10HLONG 3    ,10H
     $   ,10HAPP DIAM  ,10H          ,10HINC BIAS  ,10HSTAR ANG 2,10*10H
     $          ,10HSTAR ANG 3,10*10H          ,10HAPP DIAM  /
      MAX=60
      ITEMP=(NDIM-1)/6+1
      TRTM2=TRTM1+DELTM
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
 3000 FORMAT(1H1///5X*ERROR ANALYSIS MODE AT TRAJECTORY TIME *F12.3
     $* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8////)
      D1=TRTM1+DATEJ
      D2=TRTM2+DATEJ
      D3=D1+2415020.
      D4=D2+2415020.
      CALL TIME(D1,LYR,LMO,LDAY,LHR,LMIN,SECL,1)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE (6,3001) TRTM1,LMO,LDAY,LHR,LMIN,SECL,LYR,D3
 3001 FORMAT(/8X*INITIAL TRAJECTORY TIME*F12.5* DAYS, CALENDAR DATE*4I3,
     $F7.3,*,*I5,*, JULIAN DATE*F20.10)
      WRITE (6,3002) TRTM2,IMO,IDAY,IHR,IMIN,SECI,IYR,D4
 3002 FORMAT(/8X*FINAL TRAJECTORY TIME  *F12.5* DAYS, CALENDAR DATE*4I3,
     $F7.3,*,*I5,*, JULIAN DATE*F20.10)
      WRITE (6,3003)
 3003 FORMAT(///8X*STATE VECTOR*/27X*INITIAL*22X*FINAL*)
      WRITE (6,3004) (CMPNM(IAUG,I),XI(I),XF(I),I=1,NDIM)
 3004 FORMAT(10X,A10,E18.8,10X,E18.8)
      LINES = 18 + NDIM
      IF(LINES.LT.MAX-14) GO TO 6
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=10
```

```
6      D=TRTM1+DATEJ
       NO(1)=4
       CALL ORB(4,D)
       CALL EPHEM(1,D,1)
       DO 2  I=1,3
       RE(I)=XI(I)-XP(I)*ALNGTH
2      RE(I+3)=XI(I+3)-XP(I+3)*ALNGTH/TM
       RME=SQRT(RE(1)*RE(1)+RE(2)*RE(2)+RE(3)*RE(3))
       VME=SQRT(RE(4)*RE(4)+RE(5)*RE(5)+RE(6)*RE(6))
       NO(1)=NTP
       CALL ORB(NTP,D)
       CALL EPHEM(1,D,1)
       DO 3  I=1,3
       RTP(I)=XI(I)-XP(I)*ALNGTH
3      RTP(I+3)=XI(I+3)-XP(I+3)*ALNGTH/TM
       RMP=SQRT(RTP(1)*RTP(1)+RTP(2)*RTP(2)+RTP(3)*RTP(3))
       VMP=SQRT(RTP(4)*RTP(4)+RTP(5)*RTP(5)+RTP(6)*RTP(6))
       WRITE(6,3040) TRTM1
       WRITE(6,3041) (RE(I),I=1,3),RME,(RE(I),I=4,6),VME,(RTP(I),I=1,3),
      $RMP,(RTP(I),I=4,6),VMP
       LINES=LINES+14
       IF(LINES.LT.MAX-14) GO TO 7
       IPGN=IPGN+1
       WRITE(6,3000) TRTM2,IPROB,IPGN
       LINES=10
7      D=TRTM2+DATEJ
       NO(1)=4
       CALL ORB(4,D)
       CALL EPHEM(1,D,1)
       DO 4  I=1,3
       RE(I)=XF(I)-XP(I)*ALNGTH
4      RE(I+3)=XF(I+3)-XP(I+3)*ALNGTH/TM
       RME=SQRT(RE(1)*RE(1)+RE(2)*RE(2)+RE(3)*RE(3))
       VME=SQRT(RE(4)*RE(4)+RE(5)*RE(5)+RE(6)*RE(6))
       NO(1)=NTP
       CALL ORB(NTP,D)
       CALL EPHEM(1,D,1)
       DO 5  I=1,3
       RTP(I)=XF(I)-XP(I)*ALNGTH
5      RTP(I+3)=XF(I+3)-XP(I+3)*ALNGTH/TM
       RMP=SQRT(RTP(1)*RTP(1)+RTP(2)*RTP(2)+RTP(3)*RTP(3))
       VMP=SQRT(RTP(4)*RTP(4)+RTP(5)*RTP(5)+RTP(6)*RTP(6))
       WRITE(6,3042) TRTM2
       WRITE(6,3041) (RE(I),I=1,3),RME,(RE(I),I=4,6),VME,(RTP(I),I=1,3),
      $RMP,(RTP(I),I=4,6),VMP
       LINES=LINES+14
3040   FORMAT(///8X*AT INITIAL TRAJECTORY TIME *F12.3* DAYS*//)
3041   FORMAT(  57X*X-COMP.*13X*Y-COMP.*13X*Z-COMP.*12X*RESULTANT*//
      $10X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
      $10X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
      $10X*POSITION RELATIVE TO TARGET PLANET.  .*4E20.8/
      $10X*VELOCITY RELATIVE TO TARGET PLANET.  .*4E20.8)
3042   FORMAT(///8X  *AT FINAL TRAJECTORY TIME *F12.3* DAYS*//)
       IF(LINES.LT.MAX-6) GO TO 1
       IPGN=IPGN+1
       WRITE (6,3000) TRTM2,IPROB,IPGN
       LINES=9
1      M=MCNTR-1
       WRITE (6,3005) M
3005   FORMAT (//8X*STATISTICAL DATA AFTER MEASUREMENT*I5)
```

```
      LINES=LINES+3
      GO TO (10,20,30,40,50,60,70,80,90,100),MMCODE
10    WRITE (6,3006) TRTM2
3006  FORMAT(//10X*RANGE-RATE WAS MEASURED FROM THE IDEALIZED STATION AT
     $ TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
20    WRITE (6,3007) TRTM2
3007  FORMAT(//10X*RANGE AND RANGE-RATE WERE MEASURED FROM THE IDEALIZED
     $ STATION AT TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
30    IA=1
31    WRITE (6,3008) IA,TRTM2
3008  FORMAT (//10X*RANGE-RATE WAS MEASURED FROM STATION*I2* AT TRAJECTO
     $RY TIME*F12.5* DAYS*)
      GO TO 110
40    IA=1
41    WRITE (6,3009) IA,TRTM2
3009  FORMAT(//10X*RANGE AND RANGE-RATE WERE MEASURED FROM STATION*I2 *
     $AT TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
50    IA=2
      GO TO 31
60    IA=2
      GO TO 41
70    IA=3
      GO TO 31
80    IA=3
      GO TO 41
90    WRITE (6,3010) TRTM2
3010  FORMAT(//10X*THREE STAR PLANET ANGLES WERE MEASURED AT TRAJECTORY
     $TIME*F12.5* DAYS*)
      GO TO 110
100   WRITE(6,3011) TRTM2
3011  FORMAT(//10X*THE APPARENT PLANET DIAMETER WAS MEASURED AT TRAJECTO
     $RY TIME*F12.5* DAYS*)
110   LINES=LINES+3
      IF(LINES.LT.MAX-11) GO TO 115
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
115   WRITE (6,3012) TRTM2,TRTM1
3012  FORMAT (///10X*STATE TRANSITION MATRIX  --  PSI(*F8.3*,*F8.3*)*/)
      LINES=LINES+5
      DO 112 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 114
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
114   IF(NDIM.EQ.6) GO TO 111
      WRITE(6,3013) I
3013  FORMAT(12X*ROW *I3)
      LINES=LINES+1
111   WRITE (6,3014) (PSI(I,J),J=1,NDIM)
3014  FORMAT(18X,6E17.8)
112   LINES=LINES+ITEMP
      IF(LINES.LT.MAX-8) GO TO 113
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
113   WRITE (6,3015)
```

```
3015  FORMAT(///10X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
120   WRITE(6,3014) (Q(J,J),J=1,NDIM)
      LINES=LINES+8
      IF(LINES.LT.MAX-9) GO TO 122
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
122   WRITE (6,3016)
3016  FORMAT(///10X*OBSERVATION MATRIX*/)
      LINES=LINES+5
      DO 131 I=1,NR
      IF(LINES.LT.MAX-4) GO TO 130
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
130   IF(NDIM.EQ.6) GO TO 123
      WRITE(6,3013) I
      LINES=LINES+1
123   WRITE(6,3014) (H(I,J),J=1,NDIM)
131   LINES=LINES+ITEMP
      IF(LINES.LT.MAX-9) GO TO 132
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
132   WRITE (6,3017)
3017  FORMAT(///10X*MEASUREMENT NOISE MATRIX*/)
      DO 140 I=1,NR
140   WRITE(6,3018) (R(I,J),J=1,NR)
3018  FORMAT(18X,4E17.8)
      LINES=LINES+NR+5
      IF(LINES.LT.MAX-NDIM-5) GO TO 141
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
141   WRITE (6,3019)
3019  FORMAT(///10X*K MATRIX*/)
      DO 150 I=1,NDIM
150   WRITE (6,3018) (AK(I,J),J=1,NR)
      LINES=LINES+NDIM+5
      IF(LINES.LT.MAX-9 ) GO TO 151
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
151   WRITE (6,3020) TRTM2
3020  FORMAT(///10X          *COVARIANCE MATRIX AT TIME*F12.5* DAYS, J
     $UST BEFORE THE MEASUREMENT*/)
      LINES=LINES+5
165   DO 167 I=1,NDIM
      IF(LINES.LT.MAX-4 ) GO TO 161
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
161   IF(NDIM.EQ.6) GO TO 166
      WRITE(6,3013) I
      LINES=LINES+1
166   WRITE(6,3014) (PSIP(I,J),J=1,NDIM)
167   LINES = LINES+ITEMP
      IF(LINES.LT.MAX-9 ) GO TO 162
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
```

```
      LINES=9
162   WRITE (6,3021) TRTM2
3021  FORMAT(///10X           *COVARIANCE MATRIX AT TIME*F12.5* DAYS, A
     $FTER CONSIDERING THE MEASUREMENT*/)
      LINES=LINES+5
172   DO 174 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 171
      IPGN=IPGN+1
      WRITE (6,3000) TRTM2,IPROB,IPGN
      LINES=9
171   IF(NDIM.EQ.6) GO TO 173
      WRITE(6,3013) I
      LINES=LINES+1
173   WRITE(6,3014) (P(I,J),J=1,NDIM)
174   LINES = LINES+ITEMP
      RETURN
      END
```

```
      SUBROUTINE PRINT4(MMCODE,NR)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /MEAS/TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF
      COMMON/SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/SIM2/NB1(11),ACC1,NBOD1
      COMMON /STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON /STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON /TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION AODI(17),ADON(17),EDON(17)
      DIMENSION XE(6),XVE1(6),XVP1(6),XVE2(6),XVP2(6),XVE3(6),XVP3(6)
      F(A,B,C)=SQRT(A*A+B*B+C*C)
      MAX=60
      ITEMP=(NDIM-1)/6+1
      TRTM2=TRTM1+DELTM
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
3000  FORMAT(1H1///5X*SIMULATION MODE AT TRAJECTORY TIME*F12.3* DAYS*/
     $90X*PROBLEM. .*I10,5X*PAGE. .*I8////)
      D1=TRTM1+DATEJ
      D2=TRTM2+DATEJ
      D3=D1+2415020.
      D4=D2+2415020.
      CALL TIME(D1,LYR,LMO,LDAY,LHR,LMIN,SECL,1)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE (6,3001) TRTM1,LMO,LDAY,LHR,LMIN,SECL,LYR,D3
3001  FORMAT(/8X*INITIAL TRAJECTORY TIME*F12.5* DAYS, CALENDAR DATE*4I3,
     $F7.3,*,*I5,*, JULIAN DATE*F20.10)
      WRITE (6,3002) TRTM2,IMO,IDAY,IHR,IMIN,SECI,IYR,D4
3002  FORMAT(/8X*FINAL TRAJECTORY TIME  *F12.5* DAYS, CALENDAR DATE*4I3,
     $F7.3,*,*I5,*, JULIAN DATE*F20.10)
      WRITE (6,3003) TRTM1
3003  FORMAT(///8X*STATE VECTOR AT TIME*F8.3* DAYS*//22X*ORIGINAL NOMINA
     $L* 8X*MOST RECENT NOMINAL*12X*ACTUAL*)
      WRITE(6,3004) (CMPNM(IAUG,I),XI(I),XI1(I),ZI(I),I=1,NDIM)
      LINES=24+NDIM
      IF(LINES.LT.MAX-10-NDIM) GO TO 15
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
15    WRITE(6,3003) TRTM2
      WRITE(6,3004) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
3004  FORMAT(10X,A10,3(E20.13,5X))
      LINES=LINES+NDIM+10
      D=DATEJ+TRTM1
      ICODE=0
2     IF(LINES.LT.MAX-25) GO TO 3
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
3     NO(1)=4
```

```
      CALL ORB(4,D)
      CALL EPHEM(1,D,1)
      DO 4 I=1,3
      XE(I)=XP(I)*ALNGTH
4     XE(I+3)=XP(I+3)*ALNGTH/TM
      NO(1)=NTP
      CALL ORB(NTP,D)
      CALL EPHEM(1,D,1)
      DO 5 I=1,3
      XP(I)=XP(I)*ALNGTH
5     XP(I+3)=XP(I+3)*ALNGTH/TM
      IF(ICODE.GT.0) GO TO 7
      DO 6 I=1,6
      XVE1(I)=XI(I)-XE(I)
      XVP1(I)=XI(I)-XP(I)
      XVE2(I)=XI1(I)-XE(I)
      XVP2(I)=XI1(I)-XP(I)
      XVE3(I)=ZI(I)-XE(I)
6     XVP3(I)=ZI(I)-XP(I)
      GO TO 9
7     DO 8 I=1,6
      XVE1(I)=XF(I)-XE(I)
      XVP1(I)=XF(I)-XP(I)
      XVE2(I)=XF1(I)-XE(I)
      XVP2(I)=XF1(I)-XP(I)
      XVE3(I)=Z (I)-XE(I)
8     XVP3(I)=Z (I)-XP(I)
9     RME1=F(XVE1(1),XVE1(2),XVE1(3))
      VME1=F(XVE1(4),XVE1(5),XVE1(6))
      RMP1=F(XVP1(1),XVP1(2),XVP1(3))
      VMP1=F(XVP1(4),XVP1(5),XVP1(6))
      RME2=F(XVE2(1),XVE2(2),XVE2(3))
      VME2=F(XVE2(4),XVE2(5),XVE2(6))
      RMP2=F(XVP2(1),XVP2(2),XVP2(3))
      VMP2=F(XVP2(4),XVP2(5),XVP2(6))
      RME3=F(XVE3(1),XVE3(2),XVE3(3))
      VME3=F(XVE3(4),XVE3(5),XVE3(6))
      RMP3=F(XVP3(1),XVP3(2),XVP3(3))
      VMP3=F(XVP3(4),XVP3(5),XVP3(6))
      IF(ICODE.GT.0) GO TO 11
      WRITE(6,3026) TRTM1
3026  FORMAT(///8X*AT INITIAL TIME,*F8.3* DAYS*//59X*X-COMP.*13X*Y-COMP.
     $*13X*Z-COMP.*12X*RESULTANT*/)
      GO TO 12
11    WRITE(6,3027) TRTM2
3027  FORMAT(///8X*AT FINAL TIME *F8.3* DAYS*//59X*X-COMP.*13X*Y-COMP.*
     $13X,*Z-COMP.*12X*RESULTANT*/)
12    WRITE(6,3028) (XVE1(I),I=1,3),RME1,(XVE1(I),I=4,6),VME1,
     $(XVP1(I),I=1,3),RMP1,(XVP1(I),I=4,6),VMP1,(XVE2(I),I=1,3),RME2,
     $(XVE2(I),I=4,6),VME2,(XVP2(I),I=1,3),RMP2,(XVP2(I),I=4,6),VMP2,
     $(XVE3(I),I=1,3),RME3,(XVE3(I),I=4,6),VME3,(XVP3(I),I=1,3),RMP3,
     $(XVP3(I),I=4,6),VMP3
      LINES=LINES+25
      IF(ICODE.GT.0) GO TO 13
      IF(LINES.LT.MAX-25) GO TO 14
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
14    ICODE=1
      D=DATEJ+TRTM2
```

```
      GO TO 3
3028  FORMAT(10X*ORIGINAL NOMINAL TRAJECTORY*/
     $12X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $12X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8//
     $10X*MOST RECENT NOMINAL TRAJECTORY*/
     $12X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $12X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8//
     $10X*ACTUAL TRAJECTORY*/
     $12X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $12X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $12X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
13    IF(LINES.LT.MAX-6) GO TO 1
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
1     M=MCNTR-1
      WRITE (6,3005) M
3005  FORMAT (//8X*STATISTICAL DATA AFTER MEASUREMENT*I5)
      LINES=LINES+3
      GO TO (10,20,30,40,50,60,70,80,90,100),MMCODE
10    WRITE (6,3006) TRTM2
3006  FORMAT(//10X*RANGE-RATE WAS MEASURED FROM THE IDEALIZED STATION AT
     $ TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
20    WRITE (6,3007) TRTM2
3007  FORMAT(//10X*RANGE AND RANGE-RATE WERE MEASURED FROM THE IDEALIZED
     $ STATION AT TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
30    IA=1
31    WRITE (6,3008) IA,TRTM2
3008  FORMAT (//10X*RANGE-RATE WAS MEASURED FROM STATION*I2* AT TRAJECTO
     $RY TIME*F12.5* DAYS*)
      GO TO 110
40    IA=1
41    WRITE (6,3009) IA,TRTM2
3009  FORMAT(//10X*RANGE AND RANGE-RATE WERE MEASURED FROM STATION*I2 *
     $AT TRAJECTORY TIME*F12.5* DAYS*)
      GO TO 110
50    IA=2
      GO TO 31
60    IA=2
      GO TO 41
70    IA=3
      GO TO 31
80    IA=3
      GO TO 41
90    WRITE (6,3010) TRTM2
3010  FORMAT(//10X*THREE STAR PLANET ANGLES WERE MEASURED AT TRAJECTORY
     $TIME*F12.5* DAYS*)
      GO TO 110
100   WRITE(6,3011) TRTM2
3011  FORMAT(//10X*THE APPARENT PLANET DIAMETER WAS MEASURED AT TRAJECTO
     $RY TIME*F12.5* DAYS*)
110   LINES=LINES+3
      IF(LINES.LT.MAX-11) GO TO 115
```

```
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
115   WRITE (6,3012) TRTM2,TRTM1
3012  FORMAT (///10X*STATE TRANSITION MATRIX  --  PSI(*F8.3*,*F8.3*)*/)
      LINES=LINES+5
      DO 112 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 114
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
114   IF(NDIM.EQ.6) GO TO 111
      WRITE(6,3013) I
3013  FORMAT(12X*ROW *I3)
      LINES=LINES+1
111   WRITE (6,3014) (PSI(I,J),J=1,NDIM)
3014  FORMAT(18X,6E17.8)
112   LINES=LINES+ITEMP
      IF(LINES.LT.MAX-8) GO TO 113
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
113   WRITE (6,3015)
3015  FORMAT(///10X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
120   WRITE(6,3014) (Q(J,J),J=1,NDIM)
      LINES=LINES+8
      IF(LINES.LT.MAX-9) GO TO 122
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
122   WRITE (6,3016)
3016  FORMAT(///10X*OBSERVATION MATRIX*/)
      LINES=LINES+5
      DO 131 I=1,NR
      IF(LINES.LT.MAX-4) GO TO 130
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
130   IF(NDIM.EQ.6) GO TO 123
      WRITE(6,3013) I
      LINES=LINES+1
123   WRITE(6,3014) (H(I,J),J=1,NDIM)
131   LINES=LINES+ITEMP
      IF(LINES.LT.MAX-9) GO TO 132
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
132   WRITE (6,3017)
3017  FORMAT(///10X*MEASUREMENT NOISE MATRIX*/)
      DO 140 I=1,NR
140   WRITE(6,3018) (R(I,J),J=1,NR)
3018  FORMAT(18X,4E17.8)
      LINES=LINES+NR+5
      IF(LINES.LT.MAX-NDIM-5) GO TO 141
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
141   WRITE (6,3019)
3019  FORMAT(///10X*K MATRIX*/)
      DO 150 I=1,NDIM
150   WRITE (6,3018) (AK(I,J),J=1,NR)
```

```
      LINES=LINES+NDIM+5
      IF(LINES.LT.MAX-9 ) GO TO 151
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
151   WRITE (6,3020) TRTM2
3020  FORMAT(///10X            *COVARIANCE MATRIX AT TIME*F12.5* DAYS, J
     $UST BEFORE THE MEASUREMENT*/)
      LINES=LINES+5
165   DO 167 I=1,NDIM
      IF(LINES.LT.MAX-4 ) GO TO 161
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
161   IF(NDIM.EQ.6) GO TO 166
      WRITE(6,3013) I
      LINES=LINES+1
166   WRITE(6,3014) (PSIP(I,J),J=1,NDIM)
167   LINES = LINES+ITEMP
      IF(LINES.LT.MAX-9 ) GO TO 162
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
162   WRITE (6,3021) TRTM2
3021  FORMAT(///10X            *COVARIANCE MATRIX AT TIME*F12.5* DAYS, A
     $FTER CONSIDERING THE MEASUREMENT*/)
      LINES=LINES+5
172   DO 174 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 171
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
171   IF(NDIM.EQ.6) GO TO 173
      WRITE(6,3013) I
      LINES=LINES+1
173   WRITE(6,3014) (P(I,J),J=1,NDIM)
174   LINES = LINES+ITEMP
      IF(LINES.LT.MAX-NDIM-5) GO TO 180
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
180   WRITE(6,3022)
3022  FORMAT(///10X*ACTUAL DYNAMIC NOISE*/.
      WRITE(6,3023) (W(I),I=1,NDIM)
3023  FORMAT(18XE17.8)
      LINES=LINES+NDIM+5
      IF(LINES.LT.MAX-9) GO TO 191
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
191   WRITE(6,3034)
3034  FORMAT(///10X*MATRIX OF VARIANCES OF ACTUAL MEASUREMENT NOISE*/)
      DO 192 I=1,NR
192   WRITE(6,3018) (AR(I,J),J=1,NR)
      LINES=LINES+9
      IF(LINES.LT.MAX-9) GO TO 200
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
200   WRITE(6,3025)
```

```
3025  FORMAT(///10X*ACTUAL MEASUREMENT NOISE*/)
      WRITE(6,3023) (ANOIS(I),I=1,NR)
      LINES=LINES+9
      IF(LINES.LT.MAX-9) GO TO 210
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
210   WRITE(6,3040)
3040  FORMAT(///10X*MEASUREMENT*//23X*ESTIMATED*20X*ACTUAL*20X*RESIDUAL*
     $)
      WRITE(6,3035) (EY(I),AY(I),RES(I),I=1,NR)
3035  FORMAT(18XE17.8,10X,E17.8,10X,E17.8)
      LINES=LINES+9
      IF(LINES.LT.MAX-9) GO TO 240
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
240   WRITE(6,3030)
3030  FORMAT(///10X*RESIDUAL UNCERTAINTIES*/)
      DO 241 I=1,NR
241   WRITE(6,3018) (HPHR(I,J),J=1,NR)
      LINES=LINES+9
      DO 250 I=1,NDIM
      AODI(I)=ADEVX(I)-EDEVX(I)
      ADON(I)=XF1(I)-XF(I)+ADEVX(I)
250   EDON(I)=XF1(I)-XF(I)+EDEVX(I)
      IF(LINES.LT.MAX-NDIM-6) GO TO 190
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
190   WRITE(6,3024)
3024  FORMAT(///10X*DEVIATION OF THE STATE VECTOR FROM MOST RECENT NOMIN
     $AL* //23X*ESTIMATED*20X*ACTUAL*)
      WRITE(6,3029) (EDEVX(I),ADEVX(I),I=1,NDIM)
3029  FORMAT(18XE17.8,10XE17.8)
      LINES=LINES+NDIM+6
      IF(LINES.LT.MAX-NDIM-5) GO TO 270
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
270   WRITE(6,3032)
3032  FORMAT(///10X*DEVIATION FROM ORIGINAL NOMINAL*//23X*ESTIMATED*20X
     $*ACTUAL*)
      WRITE(6,3029) (EDON(I),ADON(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      IF(LINES.LT.MAX-NDIM-5) GO TO 260
      IPGN=IPGN+1
      WRITE(6,3000) TRTM2,IPROB,IPGN
      LINES=9
260   WRITE(6,3031)
3031  FORMAT(///10X*ACTUAL ORBIT DETERMINATION INACCURACY*/)
      WRITE(6,3023) (AODI(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      RETURN
      END
```

```
      SUBROUTINE PRNTS3(RF)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR PRINTING A SUMMARY OF THE
C     DATA GENERATED IN THE ERROR ANALYSIS MODE
C
C
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON /MEAS/ TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),VE(6),VT(6)
      F(A,B,C)=SQRT(A*A+B*B+C*C)
      MAX=60
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
1000  FORMAT(1H1///5X*S U M M A R Y   O F   E R R O R   A N A L Y S I S
     $ M O D E*//90X*PROBLEM. .*I10,5X*PAGE. .*I8////)
      DO 1 I=1,6
1     RI(I)=XB(I)
      TRTM1=TRTMB
      GO TO (10,20),NTMC
10    WRITE(6,1001)
1001  FORMAT(8X*PATCHED CONIC TRAJECTORY*///)
      GO TO 55
20    WRITE(6,2001)
2001  FORMAT(8X*VIRTUAL MASS TRAJECTORY*///)
      WRITE(6,2010) ACC,DELTH
2010  FORMAT(10X*ACCURACY FIGURE* E13.6,* INDICATES TRUE ANOMALY INCREME
     $NT IS* E20.13*  RADIANS*//)
      WRITE(6,2013) ALNGTH,TM
2013  FORMAT(10X*LENGTH UNITS*E20.10*/A.U.*/10X*TIME UNITS*E22.10*/DAY*
     $//)
      I=IEPHEM+1
      GO TO (21,22), I
21    WRITE(6,2014)
2014  FORMAT(10X*ORBITAL ELEMENTS FOR EPHEMERIS CALCULATED INITIALLY AT
     $GIVEN DATE*//)
      GO TO 23
22    WRITE(6,2015)
2015  FORMAT(10X*ORBITAL ELEMENTS FOR EPHEMERIS CALCULATED AT EVERY TIME
     $ INTERVAL*//)
23    CONTINUE
      D1=TRTMB+DATEJ
      TRTM2=FNTM
```

```
      D2 = TRTM2+DATEJ
      D3=D1+2415020.
      D4=D2+2415020.
      CALL TIME(D1,LYR,LMO,LDAY,LHR,LMIN,SECL,1)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2002) TRTMB,D3,LMO,LDAY,LHR,LMIN,SECL,LYR,TRTM2,D4,IMO,
     $IDAY,IHR,IMIN,SECI,IYR
 2002 FORMAT(10X*INITIAL TRAJECTORY TIME*F12.5* DAYS, JULIAN DATE*F20.10
     $,5X*CALENDAR DATE*4I3,F7.3,*,*I5
     $/     10X *FINAL TRAJECTORY TIME  *F12.5* DAYS, JULIAN DATE*F20.10
     $,5X*CALENDAR DATE*4I3,F7.3,*,*I5)
      RMI=F(RI(1),RI(2),RI(3))
      VMI=F(RI(4),RI(5),RI(6))
      RMF=F(RF(1),RF(2),RF(3))
      VMF=F(RF(4),RF(5),RF(6))
      WRITE(6,2003)
 2003 FORMAT(//57X*X-COMP.*13X*Y-COMP.*13X*Z-COMP.*12X*RESULTANT*)
      WRITE(6,2004) (RI(I),I=1,3),RMI,(RI(I),I=4,6),VMI,(RF(I),I=1,3),
     $RMF,(RF(I),I=4,6),VMF
 2004 FORMAT(/10X*HELIOCENTRIC ECLIPTIC COORDINATES*//
     $        10X*INITIAL POSITION OF VEHICLE . . . . .*4E20.8/
     $        10X*INITIAL VELOCITY OF VEHICLE . . . . .*4E20.8/
     $        10X*FINAL POSITION OF VEHICLE . . . . . .*4E20.8/
     $        10X*FINAL VELOCITY OF VEHICLE . . . . . .*4E20.8)
      DO 30 I=1,6
      VT(I)=RF(I)-RTP(I)
 30   VE(I)=RF(I)-RE(I)
      RMI=F(VT(1),VT(2),VT(3))
      VMI=F(VT(4),VT(5),VT(6))
      RMF=F(VE(1),VE(2),VE(3))
      VMF=F(VE(4),VE(5),VE(6))
      WRITE(6,2005) (VE(I),I=1,3),RMF,(VE(I),I=4,6),VMF,(VT(I),I=1,3),
     $RMI,(VT(I),I=4,6),VMI
 2005 FORMAT(//10X*AT FINAL TIME*//
     $10X*POSITION OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $10X*VELOCITY OF VEHICLE RELATIVE TO EARTH*4E20.8/
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      RMI=F(RC(1),RC(2),RC(3))
      VMI=F(RC(4),RC(5),RC(6))
      D4=DC+2415020.
      CALL TIME(DC,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2006) IMO,IDAY,IHR,IMIN,SECI,IYR,D4,
     $              (RC(I),I=1,3),RMI,(RC(I),I=4,6),VMI
 2006 FORMAT(//10X*AT CLOSEST APPROACH. . . . CALENDAR DATE*4I3,F7.3,*,*
     $I5,*. . .JULIAN DATE *F20.10//
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      IPGN=IPGN+1
      IF(ISPH) 40,45,40
 40   RMI=F(RSI(1),RSI(2),RSI(3))
      VMI=F(VSI(1),VSI(2),VSI(3))
      D4=DSI+2415020.
      CALL TIME(DSI,IYR,IMO,IDAY,IHR,IMIN,SECI,1)
      WRITE(6,2007) IMO,IDAY,IHR,IMIN,SECI,IYR,D4 ,RSI,RMI,VSI,VMI
 2007 FORMAT(//10X*AT SPHERE OF INFLUENCE. . . . CALENDAR DATE*4I3,F7.3,
     $*,*I5,*. . .JULIAN DATE *F20.10//
     $10X*POSITION RELATIVE TO TARGET PLANET. .*4E20.8/
     $10X*VELOCITY RELATIVE TO TARGET PLANET. .*4E20.8)
      IF(MAX.GT.51) GO TO 41
```

```
      WRITE(6,1000) IPROB,IPGN
41    WRITE(6,2008) B,BDT,BDR
2008  FORMAT(//10X*B =*E20.8*    B DOT T =*E20.8*    B DOT R =*E20.8)
      IF(MAX.LT.60) GO TO 50
      WRITE(6,1000) IPROB,IPGN
      GO TO 50
45    WRITE(6,2009)
2009  FORMAT(//10X*VEHICLE DID NOT PIERCE SPHERE OF INFLUENCE OF TARGET
     $PLANET*)
      WRITE(6,1000) IPROB,IPGN
50    CONTINUE
55    WRITE(6,3000)
3000  FORMAT(//1X130(1H*)///8X*MISCELLANEOUS DATA FOR ERROR ANALYSIS MOD
     $E*///)
      GO TO (60,70,80), ISTMC
60    WRITE(6,3001) DTMAX
3001  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY F
     $ROM THE PATCHED-CONIC TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION
     $*//15X*IF THE TIME INTERVAL OVER WHICH THE STATE TRANSITION MATRIX
     $ WAS COMPUTED WAS GREATER THAN*F8.3* DAYS*)
      IF(ISTM1) 62,61,62
61    WRITE(6,3002)
3002  FORMAT (15X*THE GOVERNING BODY WAS ASSUMED TO BE THE SUN IN THE AN
     $ALYTICAL CALCULATION*)
      GO TO 90
62    WRITE(6,3003)
3003  FORMAT(15X*THE STATE TRANSITION MATRIX CODE WAS IGNORED AND THE NU
     $MERICAL DIFFERENCING TECHNIQUE WAS USED*)
      GO TO 90
70    WRITE(6,3004) DTMAX
3004  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY F
     $ROM THE VIRTUAL MASS  TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION
     $*//15X*IF THE TIME INTERVAL OVER WHICH THE STATE TRANSITION MATRIX
     $ WAS COMPUTED WAS GREATER THAN*F8.3* DAYS*)
      IF (ISTM1) 72,71,72
71    WRITE(6,3002)
      GO TO 90
72    WRITE(6,3003)
      GO TO 90
80    WRITE(6,3005)
3005  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED FROM THE NUMER
     $ICAL DIFFERENCING TECHNIQUE*)
      WRITE(6,3006) FACP,FACV
3006  FORMAT(15X*POSITION FACTOR =*E18.10/15X*VELOCITY FACTOR =*E18.10)
      IF(NDACC) 81,90,81
81    WRITE(6,3007) ACCND
3007  FORMAT(15X*ACCURACY USED = *E18.10)
90    GO TO (100,100,100,100,91,100,100,100,91,91,100), IAUG
91    WRITE(6,3008) DELAXS,DELECC,DELICL
3008  FORMAT(/10X*WHEN THE THREE EPHEMERIS BIASES OF THE TARGET PLANET A
     $RE AUGMENTED TO THE STATE, THE FOLLOWING FACTORS WERE USED IN THE*
     ,/10X*NUMERICAL DIFFERENCING TECHNIQUE TO GENERATE THAT AUGMENTED P
     $ORTION OF THE STATE TRANSITION MATRIX*/15X*SEMI-MAJOR AXIS*E20.10/
     $15X*ECCENTRICITY   *E20.10/15X*INCLINATION   *E20.10)
100   WRITE(6,3009) NMN
3009  FORMAT(//10X*NUMBER OF MEASUREMENTS TAKEN . . . .*I5)
      WRITE(6,3010) NEV,NEV1,NEV2,NEV3
3010  FORMAT(//10X*TOTAL NUMBER OF EVENTS . . . . . . .*I5/15X*EIGENVECT
     $OR EVENTS. . . . . . .*I5/15X*PREDICTION EVENTS . . . . . . .*I5/1
     $5X*GUIDANCE EVENTS . . . . . . . .*I5)
```

```
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      IF(NEV3)110,110,101
101   WRITE(6,3011) SIGRES,SIGPRO,SIGALP,SIGBET
3011  FORMAT(10X*FOR GUIDANCE EVENTS*/15X*VARIANCE OF RESOLUTION ERROR*
     $6XE20.10/15X*VARIANCE OF PROPORTIONALITY ERROR *E20.10/15X*VARIANC
     $E OF POINTING ANGLE 1*6XE20.10/15X*VARIANCE OF POINTING ANGLE 2*6X
     $E20.10)
110   IF(NST) 120,111,120
111   WRITE(6,3012) (I,SAL(I),SLAT(I),SLON(I),I=1,NST)
3012  FORMAT(//10X*STATION LOCATION CONSTANTS*/30X*ALTITUDE*12X*LATITUDE
     $*12X*LONGITUDE*/3(15X*STATION*I2,3E20.10/))
120   IF(IDNF) 122,121,122
121   WRITE(6,3013)
3013  FORMAT(//10X*DYNAMIC NOISE IS ZERO*)
      GO TO 130
122   WRITE(6,3014) DNCN
3014  FORMAT(//10X*THE DYNAMIC NOISE MATRIX IS A DIAGONAL MATRIX WHERE T
     $HE ELEMENTS ON THE DIAGONAL ARE COMPUTED FROM THE FOLLOWING CONSTA
     $NTS*/10X6E20.10)
130   IF(IMNF)132,131,132
131   WRITE(6,3015) ((MNNAME(I,J),J=1,3),MNCN(I),I=1,12)
3015  FORMAT(//10X*MEASUREMENT NOISE WAS CONSTANT AS SHOWN BY THE FOLLOW
     $ING NUMBERS*/12(15X3A10E20.10/))
      GO TO 140
132   WRITE(6,3016)
3016  FORMAT(//10X*MEASUREMENT NOISE WAS COMPUTED INTERNALLY*)
140   I=1
      J=2
      K=3
      WRITE(6,3017) I,U1,V1,W1,J,U2,V2,W2,K,U3,V3,W3
3017  FORMAT(//10X*DIRECTION COSINES FOR THREE STAR PLANET ANGLES*/
     $3(15XI1,3E20.10/))
      WRITE(6,3018)
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      WRITE(6,3019)
3018  FORMAT(//1X130(1H*))
3019  FORMAT(8X*STATISTICAL DATA FOR ERROR ANALYSIS MODE*///)
      WRITE(6,3020) (CMPNM(IAUG,I),XB(I),XF(I),I=1,NDIM)
3020  FORMAT(10X*STATE VECTOR*/38X*INITIAL*24X*FINAL*/17(15X,A10,5XE20.1
     $0,10X,E20.10/))
      IF(NDIM.NE.6) GO TO 160
      WRITE(6,3021)
3021  FORMAT(//10X*INITIAL COVARIANCE MATRIX*/)
      DO 150 I=1,6
150   WRITE(6,3022) (PB(I,J),J=1,6)
3022  FORMAT(10X6E20.10)
      WRITE(6,3023)
3023  FORMAT(//10X*FINAL COVARIANCE MATRIX*/)
      DO 151 I=1,6
151   WRITE(6,3022) (P(I,J),J=1,6)
      GO TO 200
160   LINES = 16 + NDIM
      WRITE(6,3021)
      DO 162 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 161
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
```

```
161    WRITE(6,3024) I
3024   FORMAT(10X*ROW*I3)
       WRITE(6,3022) (PB(I,J),J=1,NDIM)
162    LINES = LINES+NDIM/6+2
       IF(LINES.LT.MAX-12) GO TO 170
       IPGN=IPGN+1
       WRITE(6,1000) IPROB,IPGN
       LINES=10
170    WRITE(6,3023)
       DO 172 I=1,NDIM
       IF(LINES.LT.MAX-4) GO TO 171
       IPGN=IPGN+1
       WRITE(6,1000) IPROB,IPGN
       LINES=10
171    WRITE(6,3024) I
       WRITE(6,3022) (P(I,J),J=1,NDIM)
172    LINES=LINES+NDIM/6+2
200    RETURN
       END
```

```
      SUBROUTINE PRNTS4(RF,RF1)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON /MEAS/ TMN(1000),MCODE(1000),NMN,MCNTR
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      DIMENSION RI(6),RF(6),RI1(6),RF1(6),RE1(6),RE2(6),RE3(6),RP1(6),
     $RP2(6),RP3(6),AODI(17),EDON(17),ADON(17)
      F(A,B,C)=SQRT(A*A+B*B+C*C)
      MAX=60
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
 1000 FORMAT(1H1///5X*S U M  MA R Y   O F   S I M U L A T I O N   M O D
     $E*/ /90X*PROBLEM. .*I10,5X,*PAGE. .*I8////)
      LINES=10
      GO TO (10,20),NTMC
 10   WRITE(6,1001)
 1001 FORMAT(8X*PATCHED CONIC TRAJECTORY*///)
      LINES=LINES+4
      GO TO 220
 20   WRITE(6,2001)
 2001 FORMAT(8X*VIRTUAL MASS TRAJECTORY*///)
      LINES=LINES+4
      WRITE(6,2002) ACC,ACC1
 2002 FORMAT(10X*ACCURACY USED IN TRAJECTORY*//14X*NOMINAL*8X*ACTUAL*/
     $12X,E10.3,5X,E10.3)
      LINES=LINES+4
      BLANK=10R
      K=NBOD1
      IF(NBOD.GT.NBOD1) K=NBOD
      WRITE(6,2003)
 2003 FORMAT(///10X*BODIES CONSIDERED IN TRAJECTORY*//12X*NOMINAL*8X*ACT
     $UAL*/)
      DO 30 I=1,K
```

```
      J1=NB(I)
      J2=NB1(I)
      IF(J1.EQ.0) GO TO 28
      IF(J2.EQ.0) GO TO 27
      WRITE(6,2004) PLANET(J1),PLANET(J2)
2004  FORMAT(12X,A10,5X,A10)
      GO TO 30
27    WRITE(6,2004) PLANET(J1),BLANK
      GO TO 30
28    IF(J2.EQ.0) GO TO 29
      WRITE(6,2004) BLANK,PLANET(J2)
      GO TO 30
29    WRITE(6,2004) BLANK,BLANK
30    CONTINUE
      LINES=LINES+K+7
      IF(LINES.LT.MAX-7 ) GO TO 40
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
40    WRITE(6,2005) DMUSB,PLANET(NTP),DMUPB
2005  FORMAT(///10X*GRAVITATIONAL CONSTANT BIASES USED IN ACTUAL TRAJECT
     $ORY*//12X*SUN*7X,E10.3/12X,A10,E10.3)
      LINES=LINES+7
      IF (LINES.LT.MAX-8) GO TO 50
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
50    WRITE(6,2006) DAB,DEB,DIB
2006  FORMAT(///10X*EPHEMERIS BIASES USED IN ACTUAL TRAJECTORY*//12X
     $*SEMI-MAJOR AXIS *E10.3/12X*ECCENTRICITY    *E10.3/12X*INCLINATION
     $     *E10.3)
      LINES=LINES+8
      D1=TRTMB+DATEJ
      D2=FNTM+DATEJ
      D3=D1+2415020.
      D4=D3+2415020.
      CALL TIME(D1,LYR,LMO,LDAY,LHR,LMIN,SECL)
      CALL TIME(D2,IYR,IMO,IDAY,IHR,IMIN,SECI)
      IF(LINES.LT.MAX-5 ) GO TO 60
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
60    WRITE(6,2020) TRTMB,D3,LMO,LDAY,LHR,LMIN,SECL,LYR,FNTM,D4,IMO,IDAY
     $,IHR,IMIN,SECI,IYR
2020  FORMAT(///10X*INITIAL TRAJECTORY TIME*F12.5* DAYS, JULIAN DATE*
     $F20.10,5X*CALENDAR DATE*4I3,F7.3*,*I5/10X*FINAL TRAJECTORY TIME  *
     $F12.5* DAYS, JULIAN DATE*F20.10,5X*CALENDAR DATE*4I3,F7.3*,*I5)
      LINES=LINES+5
      IF(LINES.LT.MAX-17) GO TO 70
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
70    WRITE(6,2007)
2007  FORMAT(///10X*AT INITIAL TIME, ECLIPTIC COORDINATES OF VEHICLE*//
     *40X*X*19X*Y*19X*Z*15X*RESULTANT*)
      RMS=F(XB(1),XB(2),XB(3))
      VMS=F(XB(4),XB(5),XB(6))
      NO(1)=4
      CALL ORB(4,D1)
      CALL EPHEM(1,D1,1)
```

```
      DO 71 I=1,3
      RE(I)=XB(I)-XP(I)*ALNGTH
71    RE(I+3)=XB(I+3)-XP(I+3)*ALNGTH/TM
      RME=F(RE(1),RE(2),RE(3))
      VME=F(RE(4),RE(5),RE(6))
      NO(1)=NTP
      CALL ORB(NTP,D2)
      CALL EPHEM(1,D2,1)
      DO 72 I=1,3
      RTP(I)=XB(I)-XP(I)*ALNGTH
72    RTP(I+3)=XB(I+3)-XP(I+3)*ALNGTH/TM
      RMP=F(RTP(1),RTP(2),RTP(3))
      VMP=F(RTP(4),RTP(5),RTP(6))
      WRITE(6,2008) (XB(I),I=1,3),RMS,(XB(I),I=4,6),VMS,(RE(I),I=1,3),
     $RME,(RE(I),I=4,6),VME,PLANET(NTP),(RTP(I),I=1,3),RMP,(RTP(I),I=4,6
     $),VMP
2008  FORMAT(14X*RELATIVE TO SUN*/16X*POSITION*5X,4E20.10/16X*VELOCITY*
     $5X,4E20.10//14X*RELATIVE TO EARTH*/16X*POSITION*5X,4E20.10/16X
     $*VELOCITY*5X,4E20.10//14X*RELATIVE TO *A10/16X*POSITION*5X,4E20.10
     $/16X*VELOCITY*5X,4E20.10)
      LINES=LINES+17
      RMS1=F(XF(1),XF(2),XF(3))
      VMS1=F(XF(4),XF(5),XF(6))
      RMS2=F(XF1(1),XF1(2),XF1(3))
      VMS2=F(XF1(4),XF1(5),XF1(6))
      RMS3=F(Z(1),Z(2),Z(3))
      VMS3=F(Z(4),Z(5),Z(6))
      NO(1)=4
      CALL ORB(4,D2)
      CALL EPHEM(1,D2,1)
      DO 80 I=1,3
      RE1(I)=XF(I)-XP(I)*ALNGTH
      RE2(I)=XF1(I)-XP(I)*ALNGTH
      RE3(I)=Z(I)-XP(I)*ALNGTH
      RE1(I+3)=XF(I+3)-XP(I+3)*ALNGTH/TM
      RE2(I+3)=XF1(I+3)-XP(I+3)*ALNGTH/TM
80    RE3(I+3)=Z(I+3)-XP(I+3)*ALNGTH/TM
      RME1=F(RE1(1),RE1(2),RE1(3))
      VME1=F(RE1(4),RE1(5),RE1(6))
      RME2=F(RE2(1),RE2(2),RE2(3))
      VME2=F(RE2(4),RE2(5),RE2(6))
      RME3=F(RE3(1),RE3(2),RE3(3))
      VME3=F(RE3(4),RE3(5),RE3(6))
      NO(1)=NTP
      CALL ORB(NTP,D2)
      CALL EPHEM(1,D2,1)
      DO 90 I=1,3
      RP1(I)=XF(I)-XP(I)*ALNGTH
      RP2(I)=XF1(I)-XP(I)*ALNGTH
      RP3(I)=Z(I)-XP(I)*ALNGTH
      RP1(I+3)=XF(I+3)-XP(I+3)*ALNGTH/TM
      RP2(I+3)=XF1(I+3)-XP(I+3)*ALNGTH/TM
90    RP3(I+3)=Z(I+3)-XP(I+3)*ALNGTH/TM
      RMP1=F(RP1(1),RP1(2),RP1(3))
      VMP1=F(RP1(4),RP1(5),RP1(6))
      RMP2=F(RP2(1),RP2(2),RP2(3))
      VMP2=F(RP2(4),RP2(5),RP2(6))
      RMP3=F(RP3(1),RP3(2),RP3(3))
      VMP3=F(RP3(4),RP3(5),RP3(6))
      IF (LINES.LT.MAX-17)  GO TO 100
```

```
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
100   WRITE(6,2009)
2009  FORMAT(///10X*AT FINAL TIME*//40X*X*19X*Y*19X*Z*15X*RESULTANT*)
      WRITE(6,2010)
2010  FORMAT(12X*ORIGINAL NOMINAL TRAJECTORY*)
      WRITE(6,2008) (XF(I),I=1,3),RMS1,(XF(I),I=4,6),VMS1,(RE1(I),I=1,3)
     $,RME1,(RE1(I),I=4,6),VME1,PLANET(NTP),
     $                          (RP1(I),I=1,3),RMP1, (RP1(I),I=4,6),VMP1
      LINES=LINES+17
      IF (LINES.LT.MAX-13) GO TO 110
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      WRITE(6,2009)
      LINES=16
      GO TO 111
110   WRITE(6,2011)
2011  FORMAT(/)
111   WRITE(6,2012)
2012  FORMAT(12X*MOST RECENT NOMINAL TRAJECTORY*)
      WRITE(6,2008) (XF1(I),I=1,3),RMS2,(XF1(I),I=4,6),VMS2,(RE2(I),I=1,
     $3),RME2,(RE2(I),I=4,6),VME2,PLANET(NTP),(RP2(I),I=1,3),RMP2,
     $(RP2(I),I=4,6),VMP2
      LINES=LINES+13
      IF(LINES.LT.MAX-13) GO TO 120
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      WRITE(6,2009)
      LINES=16
      GO TO 121
120   WRITE(6,2011)
121   WRITE(6,2013)
2013  FORMAT(12X*ACTUAL TRAJECTORY*)
      WRITE(6,2008) (Z(I),I=1,3),RMS3,(Z(I),I=4,6),VMS3,(RE3(I),I=1,3),
     $RME3,(RE3(I),I=4,6),VME3,PLANET(NTP),
     $                          (RP3(I),I=1,3),RMP3,(RP3(I),I=4,6),VMP3
      LINES=LINES+13
      RMP1=F(RCA1(1),RCA1(2),RCA1(3))
      VMP1=F(RCA1(4),RCA1(5),RCA1(6))
      RMP2=F(RCA2(1),RCA2(2),RCA2(3))
      VMP2=F(RCA2(4),RCA2(5),RCA2(6))
      RMP3=F(RCA3(1),RCA3(2),RCA3(3))
      VMP3=F(RCA3(4),RCA3(5),RCA3(6))
      IF(LINES.LT.MAX-17) GO TO 130
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
130   WRITE(6,2014) TCA1,(RCA1(I),I=1,3),RMP1,(RCA2(I),I=4,6),VMP1,TCA2,
     $(RCA2(I),I=1,3),RMP2,(RCA2(I),I=4,6),VMP2,TCA3,(RCA3(I),I=1,3),
     $RMP3,(RCA3(I),I=4,6),VMP3
2014  FORMAT(///10X*AT CLOSEST APPROACH TO THE TARGET PLANET*//40X*X*19X
     $*Y*19X*Z*15X*RESULTANT*/12X*ORIGINAL NOMINAL TRAJECTORY AT TRAJECT
     $ORY TIME*F8.3* DAYS*   /14X*POSITION*5X4E20.10/14X*VELOCITY*5X,
     $4E20.10//12X*MOST RECENT NOMINAL TRAJECTORY AT TRAJECTORY TIME*
     $F8.3* DAYS*/14X*POSITION*5X4E20.10/14X*VELOCITY*5X4E20.10//12X
     $*ACTUAL TRAJECTORY AT TRAJECTORY TIME*F8.3* DAYS*/14X*POSITION*
     $5X4E20.10/14X*VELOCITY*5X4E20.10)
      LINES=LINES+17
      IF(LINES.LT.MAX-14) GO TO 150
```

```
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
150   WRITE(6,2015)
2015  FORMAT(///10X*INFORMATION AT SPHERE OF INFLUENCE*//40X*X*19X*Y*19X
     $*Z*15X*RESULTANT*)
      WRITE(6,2010)
      IF(ISOI1.EQ.0) GO TO 160
      RMP1=F(RSOI1(1),RSOI1(2),RSOI1(3))
      VMP1=F(RSOI1(4),RSOI1(5),RSOI1(6))
      WRITE(6,2016) (RSOI1(I),I=1,3),RMP1,(RSOI1(I),I=4,6),VMP1,BSI1,
     $BDTSI1,BDRSI1,TSOI1
2016  FORMAT   (14X*POSITION*5X4E20.10/14X*VELOCITY*5X4E20.10//14X*B =*
     $E20.10*, B DOT T =*E20.10*, B DOT R =*E20.10//14X*TRAJECTORY TIME*
     $F8.3* DAYS*/)
      LINES=LINES+14
      GO TO 170
160   WRITE(6,2017) PLANET(NTP)
2017  FORMAT(14X*VEHICLE DID NOT REACH SPHERE OF INFLUENCE OF *A10/)
      LINES=LINES+9
170   IF(LINES.LT.MAX-8 ) GO TO 171
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      WRITE(6,2015)
      LINES=16
171   WRITE(6,2012)
      IF(ISOI2.EQ.0) GO TO 180
      RMP2=F(RSOI2(1),RSOI2(I),RSOI2(3))
      VMP2=F(RSOI2(4),RSOI2(5),RSOI2(6))
      WRITE(6,2016) (RSOI2(I),I=1,3),RMP2,(RSOI2(I),I=4,6),VMP2,BSI2,
     $BDTSI2,BDRSI2,TSOI2
      LINES=LINES+8
      GO TO 190
180   WRITE(6,2017) PLANET(NTP)
      LINES=LINES+3
190   IF(LINES.LT.MAX-8) GO TO 200
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      WRITE(6,2015)
      LINES=16
200   WRITE(6,2013)
      IF(ISOI3.EQ.0) GO TO 210
      RMP3=F(RSOI3(1),RSOI3(2),RSOI3(3))
      VMP3=F(RSOI3(4),RSOI3(5),RSOI3(6))
      WRITE(6,2016) (RSOI3(I),I=1,3),RMP3,(RSOI3(I),I=4,6),VMP3,BSI3,
     $BDTSI3,BDRSI3,TSOI3
      LINES=LINES+8
      LINES=LINES+3
      GO TO 220
210   WRITE(6,2017) PLANET(NTP)
      LINES=LINES+2
220   IF(LINES.LT.MAX-13) GO TO 230
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
230   WRITE(6,3000)
3000  FORMAT(//1X,130(1H*)///8X*MISCELLANEOUS INFORMATION USED IN SIMULA
     $TION MODE*///)
      LINES=LINES+9
      GO TO (240,250,260), ISTMC
```

```
240   WRITE(6,3001) DTMAX
3001  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY F
     $ROM THE PATCHED-CONIC TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION
     $*//15X*IF THE TIME INTERVAL OVER WHICH THE STATE-TRANSITION MATRIX
     $ WAS COMPUTED WAS GREATER THAN*F8.3* DAYS*)
      IF (ISTM1) 242,241,242
241   WRITE(6,3002)
3002  FORMAT (15X*THE GOVERNING BODY WAS ASSUMED TO BE THE SUN IN THE AN
     $ALYTICAL CALCULATION*)
      GO TO 270
242   WRITE(6,3003)
3003  FORMAT(15X*THE STATE TRANSITION MATRIX CODE WAS IGNORED AND THE NU
     $MERICAL DIFFERENCING CODE WAS USED*)
      GO TO 270
250   WRITE(6,3004) DTMAX
3004  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY F
     *ROM THE VIRTUAL-MASS TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION*
     $//15X*IF THE TIME INTERVAL OVER WHICH THE STATE TRANSITION MATRIX
     *WAS COMPUTED WAS GREATER THAN*F8.3* DAYS*)
      IF (ISTM1) 252,251,252
251   WRITE(6,3002)
      GO TO 270
252   WRITE(6,3003)
      GO TO 270
260   WRITE(6,3005)
3005  FORMAT(10X*THE STATE TRANSITION MATRIX WAS COMPUTED FROM THE NUMER
     $ICAL DIFFERENCING TECHNIQUE*)
      WRITE(6,3006) FACP,FACV
3006  FORMAT(15X*POSITION FACTOR =*E20.10/15X*VELOCITY FACTOR =*E20.10)
      IF(NDACC) 261,270,261
261   WRITE(6,3007) ACCND
3007  FORMAT(15X*ACCURACY USED =*E22.10)
270   LINES=LINES+4
      GO TO (290,290,290,290,280,290,290,290,280,280,290),IAUG
280   IF(LINES.LT.MAX-6) GO TO 281
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
281   WRITE(6,3008) PLANET(NTP),DELAXS,DELECC,DELICL
3008  FORMAT(/10X*THE FOLLOWING FACTORS WERE USED IN THE NUMERICAL DIFFE
     $RENCING TECHNIQUE TO GENERATE THE AUGMENTED*/10X*PORTION OF THE ST
     $ATE TRANSITION MATRIX PERTAINING TO THE EPHEMERIS BIASES OF *A10/
     $15X*SEMI-MAJOR AXIS*E20.10/15X*ECCENTRICITY   *E20.10/15X*INCLINAT
     $ION    *E20.10)
      LINES=LINES+6
290   GO TO (300,300,291,300,300,300,291,300,291,300,291), IAUG
291   IF(LINES.LT.MAX-5) GO TO 292
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
292   WRITE(6,3009) DELMUS,PLANET(NTP),DELMUP
3009  FORMAT(/10X*THE FOLLOWING FACTORS WERE USED IN THE NUMERICAL DIFFE
     $RENCING TECHNIQUE TO GENERATE THE AUGMENTED PORTION OF THG STATE*/
     $10X*TRANSITION PERTAINING TO THE BIASES OF THE GRAVITATIONAL CONST
     $ATIONAL CONSTANTS OF THE SUN AND *A10/15X*SUN*12XE20.1T/15XA10,5X,
     $E20.10)
      LINES=LINES+5
300   IF(LINES.LT.MAX-12) GO TO 301
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
```

```
      LINES=10
301   WRITE(6,3010) NMN
3010  FORMAT(///10X*NUMBER OF MEASUREMENTS TAKEN . . . .*I5)
      WRITE(6,3011) NEV,NEV1,NEV2,NEV3,NEV4
3011  FORMAT(///10X*TOTAL NUMBER OF EVENTS . . . . . . .*I5/
     $                15X*EIGENVECTOR EVENTS. . . . . . .*I5/
     $                15X*PREDICTION EVENTS . . . . . . .*I5/
     $                15X*GUIDANCE EVENTS . . . . . . . .*I5/
     $                15X*QUASI-LINEAR FILTERING EVENTS .*I5)
      LINES=LINES+12
      IF(NEV3.EQ.0) GO TO 330
      IF(LINES.LT.MAX-7 ) GO TO 310
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
310   WRITE(6,3012) SIGRES,SIGPRO,SIGALP,SIGBET
3012  FORMAT(///10X*VARIANCES OF ERRORS USED IN GUIDANCE EVENTS*//20X
     $*RESOLUTION*7X*PROPORTIONALITY*5X*POINTING ANGLE 1*4X*POINTING ANG
     $LE 2*/17XE15.8,3(5X,E15.8))
      LINES=LINES+7
      IF (LINES.LT.MAX-NEV3-6) GO TO 320
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
320   WRITE(6,3013)(I,ARES(I),APRO(I),AALP(I),ABET(I),I=1,NEV3)
3013  FORMAT(///10X*ACTUAL ERRORS USED IN GUIDANCE EVENT*//20X*RESOLUTIO
     $N*7X*PROPORTIONALITY*5X*POINTING ANGLE 1*4X*POINTING ANGLE 2*/
     $(15X,I2,E15.8,5X,E15.8,5X,E15.8,5X,E15.8))
      LINES=LINES+NEV3+6
330   IF(LINES.LT.MAX-9 ) GO TO 331
      IPGN=IPGN+1
      WRITE(6,1000)IPROB,IPGN
      LINES=10
331   WRITE(6,3014) (I,SAL(I),SLAT(I),SLON(I),I=1,3)
3014  FORMAT(///10X*STATION LOCATION CONSTANTS*/32X*ALTITUDE* 12X*LATITU
     $DE*12X*LONGITUDE*/3(15X*STATION*I2,3X,3E20.10/))
      LINES=LINES+9
      IF (LINES.LT.MAX-6) GO TO 340
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
340   IF(IDNF.GT.0) GO TO 341
      WRITE(6,3015)
3015  FORMAT(///10X*DYNAMIC NOISE IS ZERO*)
      LINES=LINES+4
      GO TO 350
341   WRITE(6,3016) DNCN
3016  FORMAT(///10X*THE DYNAMIC NOISE MATRIX IS A DIAGONAL MATRIX WHERE
     $THE DIAGONAL IS COMPUTED FROM THE FOLLOWING CONSTANTS*/26X*X*19X,
     $*Y*19X*Z*/15X3E20.10)
      LINES=LINES+6
350   IF(LINES.LT.MAX-9 ) GO TO 351
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
351   WRITE(6,3017)
3017  FORMAT(///10X*ACTUAL UNMODELLED ACCELERATION (ACTUAL DYNAMIC NOISE
     $)*//74X*X*24X*Y*24X*Z*)
      IF(FNTM.LE.TTIM1) GO TO 353
      IF(FNTM.LE.TTIM2) GO TO 352
```

```
      WRITE(6,3018) TRTMB,TTIM1,(UNMAC(I,1),I=1,3)
      WRITE(6,3018) TTIM1,TTIM2,(UNMAC(I,2),I=1,3)
      WRITE(6,3018) TTIM2,FNTM ,(UNMAC(I,3),I=1,3)
 3018 FORMAT(15X*FROM*F8.3* DAYS THROUGH *F8.3* DAYS . . .*3E25.13)
      LINES=LINES+9
      GO TO 360
  352 WRITE(6,3018) TRTMB,TTIM1,(UNMAC(I,1),I=1,3)
      WRITE(6,3018) TTIM1,FNTM ,(UNMAC(I,2),I=1,3)
      LINES=LINES+8
      GO TO 360
  353 WRITE(6,3018) TRTMB,FNTM ,(UNMAC(I,1),I=1,3)
      LINES=LINES+7
  360 IF(LINES.LT.MAX-16) GO TO 361
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
  361 IF(IMNF) 363,362,363
  362 WRITE(6,3019) ((MNNAME(I,J),J=1,3),MNCN(I),I=1,12)
 3019 FORMAT(///10X*MEASUREMENT NOISE WAS CONSTANT*/12(15X3A10,E20.10/))
      LINES=LINES+16
      GO TO 370
  363 WRITE(6,3020)
 3020 FORMAT(///10X*MEASUREMENT NOISE WAS COMPUTED INTERNALLY*)
      LINES=LINES+4
  370 IF (LINES.LT.MAX-17) GO TO 371
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
  371 IF(IAMNF.GT.0) GO TO 372
      WRITE(6,3021)
 3021 FORMAT(///10X*THE UNCERTAINTIES IN THE ACTUAL MEASUREMENT NOISE WE
     $RE ASSUMED TO BE THE SAME AS*/10X*THE UNCERTAINTIES IN THE MEASURE
     $MENT NOISE OF THE MOST RECENT NOMINAL*)
      LINES=LINES+5
      GO TO 380
  372 WRITE(6,3022) ((MNNAME(I,J),J=1,3),AVARM(I),I=1,12)
 3022 FORMAT(///10X*THE UNCERTAINTIES IN THE ACTUAL MEASUREMENTS ARE COM
     $PUTED FROM THE FOLLOWING VARIANCES* /12(15X3A10,E20.10/))
      LINES=LINES+17
  380 I=1
      J=2
      K=3
      IF(LINES.LT.MAX-8 ) GO TO 381
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
  381 WRITE(6,3023) I,U1,V1,W1,J,U2,V2,W2,K,U3,V3,W3
 3023 FORMAT(///10X*DIRECTION COSINES FOR THREE STAR PLANET ANGLES*/
     $3(15X,I1,3E20.10/))
      LINES=LINES+8
      IF(LINES.LT.MAX-NDIM-12) GO TO 390
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
  390 WRITE(6,3024)
 3024 FORMAT(///8X*STATISTICAL DATA FOR SIMULATION MODE*/)
      LINES=LINES+5
      DO 391 I=1,NDIM
  391 ZI(I)=XB(I)+ADEVXB(I)
      WRITE (6,3025) (CMPNM(IAUG,I),XB(I),ZI(I),I=1,NDIM)
```

```
3025  FORMAT(///10X*INITIAL STATE VECTOR*//33X*NOMINAL*14X*ACTUAL*/
     $17(15X,A10,E20.10,E20.10/))
      LINES=LINES+NDIM+7
      IF(LINES.LT.MAX-NDIM-7) GO TO 400
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
400   WRITE(6,3026) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
3026  FORMAT(///10X*FINAL STATE VECTOR*//29X*ORIGINAL NOMINAL*2X*MOST RE
     $CENT NOMINAL*8X*ACTUAL*/17(15X,A10,3E20.10/))
      LINES=LINES+NDIM+7
      IF(LINES.LT.MAX-NDIM-7) GO  TO 410
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
410   WRITE(6,3031) (EDEVX(I),ADEVX(I),I=1,NDIM)
3031  FORMAT(///10X*DEVIATION OF THE STATE VECTOR FROM THE MOST RECENT N
     $OMINAL TRAJECTORY AT FINAL TIME*//22X*ESTIMATED*13X*ACTUAL*/17(15X
     $2E20.10/))
      LINES=LINES+NDIM+7
      IF(LINES.LT.MAX-NDIM-7) GO TO 411
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
411   DO 412 I=1,NDIM
      AODI(I)=ADEVX(I)-EDEVX(I)
      ADON(I)=XF1(I)-XF(I)+ADEVX(I)
412   EDON(I)=XF1(I)-XF(I)+EDEVX(I)
      WRITE(6,3032) (EDON(I),ADON(I),I=1,NDIM)
3032  FORMAT(///10X*DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL AT
     $FINAL TIME*//22X*ESTIMATED*13X*ACTUAL*/17(15X2E20.10/))
      LINES=LINES+NDIM+7
      IF (LINES.LT.MAX-NDIM-6) GO TO 413
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
413   WRITE(6,3033) (AODI(I),I=1,NDIM)
3033  FORMAT(///10X*ACTUAL ORBIT DETERMINATION INACCURACY AT FINAL TIME
     $*//17(15XE20.10/))
      LINES=LINES+NDIM+6
      IF(LINES.LT.MAX-9) GO TO 420
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
420   WRITE(6,3027)
3027  FORMAT(///10X*INITIAL COVARIANCE MATRIX*/)
      LINES=LINES+5
      DO 423 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 421
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
421   IF(NDIM.EQ.6) GO TO 422
      WRITE(6,3028) I
3028  FORMAT(12X*ROW*I3)
      LINES=LINES+1
422   WRITE(6,3029) (PB(I,J),J=1,NDIM)
3029  FORMAT(15X6E18.8)
423   LINES=LINES+(NDIM-1)/6+1
      IF(LINES.LT.MAX-9) GO TO 430
```

```
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
430   WRITE(6,3030)
3030  FORMAT(///10X*FINAL COVARIANCE MATRIX*/)
      LINES=LINES+5
      DO 433 I=1,NDIM
      IF(LINES.LT.MAX-4) GO TO 431
      IPGN=IPGN+1
      WRITE(6,1000) IPROB,IPGN
      LINES=10
431   IF(NDIM.EQ.6) GO TO 432
      WRITE(6,3028) I
      LINES=LINES+1
432   WRITE(6,3029) (P(I,J),J=1,NDIM)
433   LINES=LINES+(NDIM-1)/6+1
      RETURN
      END
```

```
      SUBROUTINE PSIM(RI,RF,ISC)
C THIS STATE TRANSITION MATRIX MODULE CHECKS THE CODE TO DETERMINE
C HOW THE STM ARE TO BE COMPUTED
C   1 = PATCHED CONIC
C   2 = VIRTUAL MASS
C   3 = NUMERICAL DIFFERENCING
C
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DLETP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RF(6),RS(3),VS(3),DUM(6,6),VEC(6)
      DO 1 I=1,NDIM
      DO 1 J=1,NDIM
1     PSI(I,J)=0.
      IF(ISC.EQ.3) GO TO 30
      IF(DELTM.LE.DTMAX) GO TO 5
      IF(ISTM1.NE.0) GO TO 30
      DO 2 I=1,3
      RS(I)=RI(I)
2     VS(I)=RI(I+3)
      VMU=PMASS(1)*ALNGTH*ALNGTH*ALNGTH/(TM*TM)
      GO TO 23
5     GO TO (10,20,30), ISC
10    CALL PCTM(RI)
      GO TO 40
20    DO 21 I=1,3
      RS(I)=RVS(I)
21    VS(I)=RVS(I+3)
23    DELT=DELTM*TM
      CALL CONC2(RS,VS,DELT,VMU,DUM)
      DO 22 I=1,6
      DO 22 J=1,6
22    PSI(I,J)=DUM(I,J)
      GO TO 40
30    CALL NDTM(RI,RF)
40    IF(IAUG.EQ.1) GOTO 100
      THSP=6.*(SPHERE(NTP)*ALNGTH)
      D=DATEJ+TRTM1
      NO(1)=NTP
      CALL ORB(NTP,D)
      CALL EPHEM(1,D,1)
      DO 41 I=1,3
41    VEC(I)=RI(I)-XP(I)*ALNGTH
      POSS=SQRT(VEC(1)*VEC(1)+VEC(2)*VEC(2)+VEC(3)*VEC(3))
      GO TO (100,50,60,50,70,50,60,50,80,70,60), IAUG
50    DO 51 I=7,NDIM
51    PSI(I,I)=1.
      GO TO 100
60    CALL MUND(RI,RF,POSS)
      GO TO 50
```

```
70     IF(POSS.LE.THSP) CALL PLND(RI,RF)
       GO TO 50
80     CALL MUND(RI,RF,POSS)
       GO TO 70
       GO TO 50
100    RETURN
       END
```

```
      SUBROUTINE QUASI(RI,TEVN,RI1)
C     THIS ROUTINE CONTAINS THE LOGIC FOR THE QUASI-LINEAR FILTERING
C     EVENT IN THE SIMULATION MODE OF THE STEAP PROGRAM
C
C     ARGUMENTS ARE DEFINED BELOW
C        RI     --  POSITION AND VELOCITY ON ORIGINAL NOMINAL TRAJECTORY
C                   AT TIME TEVN
C        TEVN   --  TRAJECTORY TIME OF QUASI-LINEAR FILTERING EVENT
C        RI1    --  POSITION AND VELOCITY ON MOST RECENT NOMINAL
C                   TRAJECTORY
C
C
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON /CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON /EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON /SIM2/NB1(11),ACC1,NBOD1
      COMMON /STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON /STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON /TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI(6),RI1(6),RF(6),RF1(6),RI2(6),RF2(6),DUM(17)
      DIMENSION RHO(17,17)
      MAX=60
      DELTM=TEVN-TRTM1
      CALL NTM(RI,RF,NTMC,1)
      DO 10 I=1,6
10    XF(I)=RF(I)
      IF (NQE.NE.0) GO TO 20
      DO 11 I=1,NDIM
11    XF1(I)=XF(I)
      DO 12 I=1,6
12    RF1(I)=RF(I)
      GO TO 30
20    CALL NTM(RI1,RF1,NTMC,2)
      DO 21 I=1,6
21    XF1(I)=RF1(I)
30    CALL PSIM(RI1,RF1,ISTMC)
      NQE=NQE+1
      CALL DYNO(0)
      CALL NAVM(1,1)
      DO 40 I=1,NDIM
      DO 40 J=I,NDIM
      RHO(I,J)=P(I,J)/SQRT(P(I,I)*P(J,J))
40    RHO(J,I)=RHO(I,J)
      DO 50 I=1,6
50    RI2(I)=XI1(I)+ADEVX(I)
```

```
      CALL NTM(RI2,RF2,NTMC,3)
      DO 51 I=1,6
51    Z(I)=RF2(I)
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      WRITE(6,3001)
      LINES=12
      WRITE(6,3002) (CMPNM(IAUG,I),XF(I),XF1(I),Z(I),I=1,NDIM)
      LINES=LINES+NDIM
      WRITE(6,3004) TEVN,TRTM1
      LINES=LINES+5
      DO 33 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 31
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
31    IF(NDIM.EQ.6) GO TO 32
      WRITE(6,3013) I
      LINES=LINES+1
32    WRITE(6,3014) (PSI(I,J),J=1,NDIM)
33    LINES=LINES+(NDIM-1)/6+1
      IF (LINES.LT.MAX-8) GO TO 34
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
34    WRITE (6,3003)
      WRITE (6,3014) (Q(I,I),I=1,NDIM)
      LINES=LINES+8
      IF (LINES.LT.MAX-9) GO TO 35
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
35    WRITE(6,3005) TEVN,TRTM1
      LINES=LINES+5
      DO 38 I=1,NDIM
      IF (LINES.LT.MAX-4) GO TO 36
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
36    IF(NDIM.EQ.6) GO TO 37
      WRITE(6,3013) I
      LINES=LINES+1
37    WRITE(6,3014) (P(I,J),J=1,NDIM)
38    LINES=LINES+(NDIM-1)/6+1
      IF(LINES.LT.MAX-9) GO TO 41
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
41    WRITE(6,3006) TEVN
      DO 44 I=1,NDIM
      IF (LINES.LT.MAX-4)GO TO 42
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
42    IF(NDIM.EQ.6) GO TO 43
      WRITE(6,3013) I
      LINES=LINES+1
43    WRITE(6,3014) (RHO(I,J),J=1,NDIM)
44    LINES=LINES+(NDIM-1)/6+1
      CALL DYNO(1)
```

```
      IF (LINES.LT.MAX-NDIM-5) GO TO 53
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
53    WRITE(6,3008) (W(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      DO 60 I=1,6
60    ADEVX(I)=Z(I)+W(I)-XF1(I)
      DO 70  I=1,NDIM
      DUM(I)=0.
      DO 70  J=1,NDIM
70    DUM(I)=DUM(I)+PSI(I,J)*EDEVX(J)
      DO 71  I=1,NDIM
71    EDEVX(I)=DUM(I)
      IF (LINES.LT.MAX-NDIM-7) GO TO 72
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
72    WRITE(6,3010) (EDEVX(I),ADEVX(I),I=1,NDIM)
      LINES=LINES+NDIM+7
      DO 83 I=1,6
83    XF1(I)=XF1(I)+EDEVX(I)
      DO 80 I=1,NDIM
      XI1(I)=XF1(I)
80    XI(I)=XF(I)
      DO 90 I=1,6
      ADEVX(I)=ADEVX(I)-EDEVX(I)
90    EDEVX(I)=0.
      IF (LINES.LT.MAX-NDIM-5) GO TO 81
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
81    WRITE (6,3011)(XI1(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      IF(LINES.LT.MAX-5-NDIM) GO TO 82
      IPGN=IPGN+1
      WRITE(6,3000) (MDNM(ITR,K),K=1,2),TEVN,IPROB,IPGN
      LINES=9
82    WRITE (6,3012) (ADEVX(I),I=1,NDIM)
      LINES=LINES+NDIM+5
      TRTM1=TEVN
      RETURN
3000  FORMAT(1H1//8X2A10*--QUASILINEAR FILTERING EVENT AT TRAJECTORY TIM
     $E *F12.3* DAYS*/90X*PROBLEM. .*I10,5X*PAGE. .*I8///1X,130(1H*)/)
3001  FORMAT(///8X*STATE VECTOR*//22X*ORIGINAL NOMINAL*7X*MOST RECENT NO
     $MINAL*13X*ACTUAL*)
3002  FORMAT(8X,A10,E20.10,5X,E20.10,5X,E20.10)
3003  FORMAT(///8X*DIAGONAL OF DYNAMIC NOISE MATRIX*/)
3004  FORMAT(///8X*STATE TRANSITION MATRIX RELATING THE STATE VECTOR AT
     $TIME *F8.3* DAYS TO THAT AT TIME *F8.3* DAYS*/)
3005  FORMAT(///8X*COVARIANCE MATRIX AT TIME OF QUASI-LINEAR FILTERING E
     $VENT  --  P(*F8.3*,*F8.3*)*/)
3006  FORMAT(///8X*CORRELATION COEFFICIENT MATRIX AT TIME *F8.3* DAYS*/)
3008  FORMAT(///8X*ACTUAL DYNAMIC NOISE*//(8XE20.10))
3010  FORMAT(///8X*DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NO
     $MINAL TRAJECTORY*//15X*ESTIMATED*13X*ACTUAL*/(8X2E20.10))
3011  FORMAT(///8X*STATE VECTOR OF NEW NOMINAL TRAJECTORY*//(8XE20.10))
3012  FORMAT(///8X*ACTUAL DEVIATION OF NEW STATE VECTOR*//(8XE20.10))
3013  FORMAT(10X*ROW*I3)
3014  FORMAT(10X6E20.10)
      END
```

```
      FUNCTION RNUM(SIGMA)
C     THIS FUNCTION GENERATES RANDOM VARIABLES FROM A NORMAL
C     DISTRIBUTION WITH MEAN ZERO AND STANDARD DEVIATION SIGMA.
      DATA NX/0/
      A=0.0
      DO 100 I=1,12
      IF(NX)3,2,3
    2 YY=5160736.
      ZZ=1492480.
      WW=3130862.
      SS=6538271.
      NX = 2
    3 WW= WW+WW
      YY= YY+YY
      ZZ= ZZ+ZZ
      Y1 =YY-9999997.
      Z1 =ZZ-9999971.
      W1 =WW-9699691.
      IF(Y1)20,20,10
   10 YY = Y1
   20 IF (Z1) 40,40,30
   30 ZZ=Z1
   40 IF(W1) 60,60,50
   50 WW =W1
   60 SS=WW+ZZ+YY+SS
      N=SS*.0000001
      Q=N*10000000
      SS =SS-Q
      RR=SS*.0000001
100   A=A+RR
      RNUM=(A-6.)*SIGMA
      RETURN
      END
```

```
      SUBROUTINE SCHED(T1,T2,MMCODE)
      COMMON /MEAS/ TMN(1000),MCODE(1000),NMN,MCNTR
5     IF(MCNTR-NMN) 10,10,30
10    DO 15 M=MCNTR,NMN
      IF(T1-TMN(M)) 20,20,15
15    CONTINUE
20    T2=TMN(M)
      MMCODE=MCODE(M)
30    RETURN
      END
```

```
      SUBROUTINE SPACE (LINES)
C
C
C
C     THIS SUBROUTINE COUNTS THE NUMBER OF LINES BEING PRINTED TO
C     DETERMINE WHEN TO SKIP TO THE NEXT PAGE WITH A NEW HEADING.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      IF (LINPGE.LT.(LINCT+LINES)) CALL NEWPGE
      LINCT=LINCT+LINES
      RETURN
      END
```

```
      SUBROUTINE STAPARL(AL,ALON,ALAT,PAT2,VEC,PA)
C
C     THIS SUBROUTINE COMPUTES THE PARTIAL DERIVATIOVES FOR STATION
C     LOCATION ERRORS
C
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(6),MNCN(12)
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION VEC(6),PA(6,3)
      G1=SIN(ALAT)
      G2=COS(ALAT)
      G3=SIN(PAT2)
      G4=COS(PAT2)
      G5=SIN(EPS)
      G6=COS(EPS)
C
      PA(1,1)=-G2*G4
      PA(1,2)=AL*G1*G4
      PA(1,3)=AL*G2*G3
      PA(2,1)=-(G5*G1+G6*G2*G3)
      PA(2,2)=AL*G6*G1*G3-AL*G6*G2
      PA(2,3)=-AL*G6*G2*G4
      PA(3,1)=G5*G2*G3-G6*G1
      PA(3,2)=-(AL*G5*G1*G3+AL*G6*G2)
      PA(3,3)=AL*G5*G2*G4
      OMEG=OMEGA/TM
      PA(4,1)=OMEG *G2*G3
      PA(4,2)=(-OMEG )*AL*G1*G3
      PA(4,3)=OMEG *G2*G4*AL
      PA(5,1)=(-OMEG )*G2*G4*G6
      PA(5,2)=OMEG *G6*G1*G4*AL
      PA(5,3)=OMEG *AL*G6*G2*G3
      PA(6,1)=OMEG *G5*G2*G4
      PA(6,2)=(-OMEG )*AL*G5*G1*G4
      PA(6,3)=(-OMEG )*AL*G5*G2*G3
      RETURN
      END
```

```
      SUBROUTINE TIME(DAY,IYR,MO,IDAY,IHR,MIN,SEC,ICODE)
C
C
C
C     THIS SUBROUTINE CHOOSES BETWEEN TWO OPTIONS.
C
C          (1)  CONVERTS FORM CALENDAR DATE TO JULIAN DATE, EPOCH
C               JAN. 0, 1900, OR
C          (2)  CONVERTS FROM JULIAN DATE, EPOCH JAN. 0, 1900, TO
C               CALENDAR DATE, DEPENDING ON THE VARIABLE ICODE.
C
C               ICODE.EQ.0  OPTION 1
C               ICODE.NE.0  OPTION 2
C               DAY - FLOATING POINT JULIAN DATE, EPOCH JAN. 0, 1900
C
C
C
      IF (ICODE) 10,50,10
 10   P=DAY+2415020.5
      JD=P
      R=P-FLOAT(JD)
      JD=JD-1721119
      J=4*JD-1
      IT=J/146097
      J=(J-146097*IT)/4
      J=4*J+3
      IV=J/1461
      J=(J+4-1461*IV)/4
      J=5*J
      IX=(J-3)/153
      IDAY    =(J+2-153*IX)/5
      IF(IX-10)20,30,30
 20   MO      =IX+3
      IYR     =100*IT+IV
      GO TO 40
 30   MO      =IX-9
      IYR     =100*IT+IV+1
 40   R=24.*R
      IHR     =R
      R=60.*(R-FLOAT(IHR))
      MIN = R
      R=60.*(R-FLOAT(MIN))
      SEC = R
      RETURN
 50   IF(MO-2)60,60,70
 60   IP=MO + 9
      IQ=IYR - 1
      GO TO 80
 70   IP=MO - 3
      IQ=IYR
 80   IA=IQ/100
      IB=IQ-100*IA
      IR=IA/4
      IS=IA-4*IR
      IT=IB/4
      IU=IB-4*IT
      IV=(153*IP+2)/5
      DAY=146097*IR+36524*IS+1461*IT+365*IU+IV+IDAY-693901
      DAY=DAY-.5+FLOAT(IHR) /24.  +FLOAT(MIN) /1440.  +
     1SEC/86400.
```

```
      RETURN
      END
```

```
      SUBROUTINE TRAKM(HECV,ITRK,NR,IOBS,VECTOR)
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON/CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/CONST3/DELXA,DELYA,DELZA,DELXE,DELYE,DELZE,DELXI,DELYI,
     $DELZI,DELAXS,DELECC,DELICL,DELMUS,DELMUP
      COMMON/SIMCNT/DMUSB,DMUPB,DAB,DEB,DIB,TTIM1,TTIM2,UNMAC(3,3),
     $SLB(9),AVARM(12),IAMNF,ARES(20),APRO(20),AALP(20),ABET(20)
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT ,TRTMB
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION HECV(6),VEC(6),GECS(6),GELS(6),PA(6,3),HECP(6),HECE(6)
      DIMENSION VECTOR(4)
C
C
C     THIS SUBROUTINE COMPUTES THE MEASUREMENT MATRIX H FOR AUGMENTED
C     AND NON-AUGMENTED STATES, THE ARGUMENTS ARE
C
C     HECV -- HELIOCENTRIC ECLIPTIC COORDINATES OF VEHICLE
C     ITRK -- CODE TO DETERMINE WHICH TRACKING MODEL WILL BE USED
C     OUTPUT QUANTITIES ARE
C
C     H    -- MEASUREMENT MATRIX
C     NR   -- NUMBER OF ROWS IN H
C
C
      NO(1)=4
      D=DATEJ+TRTM1+DELTM
      CALL ORB(4,D)
      CALL EPHEM(1,D,1)
      DO 300 I=1,3
      HECE(I)=XP(I)*ALNGTH
300   HECE(I+3)=XP(I+3)*ALNGTH/TM
      NO(1)=NTP
      CALL ORB(NTP,D)
      CALL EPHEM(1,D,1)
      DO 301 I=1,3
      HECP(I)=XP(I)*ALNGTH
301   HECP(I+3)=XP(I+3)*ALNGTH/TM
      T=DATEJ-18262.5+TRTM1+DELTM
      IF(IOBS.NE.0) GO TO 302
      DO 2 I=1,4
      DO 2 J=1,17
      H(I,J) = 0.0
    2 CONTINUE
302   GO TO (1,1,3,3,4,4,5,5,6,7),ITRK
    1 DO 100 IN=1,6
  100 VEC(IN) = HECV(IN) - HECE(IN)
      R1= SQRT(VEC(1)*VEC(1) + VEC(2)*VEC(2) + VEC(3)*VEC(3))
      RRATE =(VEC(1)*VEC(4)  + VEC(2)*VEC(5) + VEC(3)*VEC(6)) /R1
      IF(IOBS.EQ.0) GO TO 400
      IF(ITRK.EQ.1) GO TO 401
      VECTOR(1)=R1
      VECTOR(2)=RRATE
      NR=2
```

```
      GO TO 200
401   VECTOR(1)=RRATE
      NR=1
      GO TO 200
400   A1 = VEC(1) /R1
      A2 = VEC(2) /R1
      A3 = VEC(3) /R1
      R2=R1*R1
      B1 = VEC(4)/R1- (VEC(1)*RRATE)/R2
      B2 = VEC(5)/R1- (VEC(2)*RRATE)/R2
      B3 = VEC(6)/R1- (VEC(3)*RRATE)/R2
      GO TO (15,25),ITRK
   15 H(1,1) = B1
      H(1,2) = B2
      H(1,3) = B3
      H(1,4) = A1
      H(1,5) = A2
      H(1,6) = A3
      NR = 1
      GO TO 200
   25 H(1,1) = A1
      H(1,2) = A2
      H(1,3) = A3
      H(2,1) = B1
      H(2,2) = B2
      H(2,3) = B3
      H(2,4) = A1
      H(2,5) = A2
      H(2,6) = A3
      NR = 2
      GO TO 200
    3 IA = 1
      GO TO 12
    4 IA = 2
      GO TO 12
    5 IA = 3
   12 AL = SAL(IA) + RADIUS(4)*ALNGTH
      ALON = SLON(IA)
      ALAT = SLAT(IA)
      IF(IOBS.NE.2) GO TO 13
      AL=AL+SLB(3*IA-2)
      ALAT=ALAT+SLB(3*IA-1)
      ALON=ALON+SLB(3*IA)
13    PAT1 = AL*COS(ALAT)
      PAT2 = ALON + OMEGA *(T-UNIVT)
      CP=COS(PAT2)
      SP=SIN(PAT2)
      GECS(1) = PAT1*CP
      GECS(2) = PAT1*SP
      GECS(3) = AL*SIN(ALAT)
      GECS(4) = (-OMEGA)*PAT1*SP/TM
      GECS(5) = OMEGA*PAT1*CP/TM
      GECS(6) = 0.0
      CE=COS(EPS)
      SE=SIN(EPS)
      GELS(1) = GECS(1)
      GELS(2) = GECS(2)*CE     + GECS(3)*SE
      GELS(3) = (-GECS(2))*SE     + GECS(3)*CE
      GELS(4) = GECS(4)
      GELS(5) = GECS(5)*CE
```

```
      GELS(6)=-GECS(5)*SE
      DO 30 I =1,6
      VEC(I) = HECV(I) - HECE(I) - GELS(I)
   30 CONTINUE
      R1= SQRT(VEC(1)*VEC(1) + VEC(2)*VEC(2) + VEC(3)*VEC(3))
      RRATE = (VEC(1)*VEC(4) + VEC(2)*VEC(5) + VEC(3)*VEC(6)) /R1
      IF(IOBS.EQ.0) GO TO 402
      IF(ITRK/2*2.NE.ITRK) GO TO 403
      VECTOR(1)=R1
      VECTOR(2)=RRATE
      NR=2
      GO TO 200
403   VECTOR(1)=RRATE
      NR=1
      GO TO 200
402   A1 = VEC(1)/R1
      A2 = VEC(2)/R1
      A3 = VEC(3)/R1
      R2=R1*R1
      B1 = VEC(4)/R1- (VEC(1)*RRATE)/R2
      B2 = VEC(5)/R1- (VEC(2)*RRATE)/R2
      B3 = VEC(6)/R1- (VEC(3)*RRATE)/R2
   35 H(1,1) = B1
      H(1,2) = B2
      H(1,3) = B3
      H(1,4) = A1
      H(1,5) = A2
      H(1,6) = A3
      IF(ITRK-4) 50,50,55
55    IF(IAUG-6) 60,61,60
60    IF(ITRK/2*2.EQ.ITRK) GO TO 40
      NR = 1
      GO TO 200
61    CALL STAPARL(AL,ALON,ALAT,PAT2,VEC,PA)
      E1=PA(1,1)*B1  +  PA(2,1)*B2  +  PA(3,1)*B3  +
     *(PA(4,1)*VEC(1) + PA(5,1)*VEC(2) + PA(6,1)* VEC(3))/R1
      E2 = PA(1,2)*B1  +  PA(2,2)*B2  +  PA(3,2)*B3  +
     *(PA(4,2)*VEC(1) + PA(5,2)*VEC(2) + PA(6,2)*VEC(3))/R1
      E3 = PA(1,3)*B1  +  PA(2,3)*B2  +  PA(3,3)*B3  +
     *(PA(4,3)*VEC(1) + PA(5,3)*VEC(2) + PA(6,3)*VEC(3))/R1
      IF(ITRK.GE. 7) GO TO 62
      IF(ITRK.LE.4) GO TO 53
      H(1,10) = E1
      H(1,11) = E2
      H(1,12)=E3
      GO TO 60
62    H(1,13)=E1
      H(1,14)=E2
      H(1,15)=E3
      GO TO 60
50    GO TO(60,61,60,52,60,61,61,61,61,60,52,61), IAUG
   52 H(1,8) = 1.0
      GO TO 60
53    H(1,7)=E1
      H(1,8)=E2
      H(1,9)=E3
      IF(IAUG .EQ.  8) GO TO 54
      IF(IAUG .EQ. 11) GO TO 56
      GO TO 60
   54 H(1,11) = 1.0
```

```
      GO TO 60
   56 H(1,13) = 1.0
      GO TO 60
40    DO 41 I=1,NDIM
      H(2,I)=H(1,I)
41    H(1,I)=0.
      H(1,1)=A1
      H(1,2) = A2
      H(1,3) = A3
      E1       =(PA(1,1)*VEC(1) + PA(2,1)*VEC(2) + PA(3,1)*VEC(3))/R1
      E2       =(PA(1,2)*VEC(1) + PA(2,2)*VEC(2) + PA(3,2)*VEC(3))/R1
      E3       =(PA(1,3)*VEC(1) + PA(2,3)*VEC(2) + PA(3,3)*VEC(3))/R1
      NR=2
      IF(ITRK-6) 42,47,48
42    GO TO (200,43,200,46,200,43,43,43,200,46,43),IAUG
43    H(1,7)=E1
      H(1,8)=E2
      H(1,9)=E3
      IF(IAUG.NE.8) GO TO 44
      H(1,10)=1.0
      GO TO 200
44    IF(IAUG.NE.11) GO TO 200
      H(1,12)=1.0
      GO TO 200
46    H(1,7)=1.0
      GO TO 200
47    IF(IAUG.NE.6) GO TO 200
      H(1,10)=E1
      H(1,11) = E2
      H(1,12)=E3
      GO TO 200
48    IF(IAUG.NE.6) GO TO 200
      H(1,13)=E1
      H(1,14)=E2
      H(1,15)=E3
      GO TO 200
    6 DO 80 J = 1,6
      VEC(J) = HECP(J) - HECV(J)
   80 CONTINUE
      RHO = SQRT(VEC(1)*VEC(1) +  VEC(2)*VEC(2) + VEC(3)*VEC(3))
      RHO2=RHO*RHO
      COAL1 =           (U1*VEC(1) + V1*VEC(2) + W1*VEC(3))/RHO
      COAL2 =           (U2*VEC(1) + V2*VEC(2) + W2*VEC(3))/RHO
      COAL3 =           (U3*VEC(1) + V3*VEC(2) + W3*VEC(3))/RHO
      SIAL1 = SQRT(1.0  - COAL1 * COAL1)
      SIAL2 = SQRT(1.0  - COAL2 * COAL2)
      SIAL3 = SQRT(1.0  - COAL3 * COAL3)
      IF(IOBS.EQ.0) GO TO 87
      VECTOR(1)=ASIN(SIAL1)
      VECTOR(2)=ASIN(SIAL2)
      VECTOR(3)=ASIN(SIAL3)
      GO TO 200
87    IF(ABS(SIAL1) .GE. .001) GO TO 81
      S11=(1000.0)*(U1/RHO - VEC(1)/RHO2)
      S12=(1000.0)*(V1/RHO - VEC(2)/RHO2)
      S13=(1000.0)*(W1/RHO - VEC(3)/RHO2)
      GO TO 82
   81 S11=              (U1/RHO - (VEC(1)*COAL1)/RHO2)/SIAL1
      S12=              (V1/RHO - (VEC(2)*COAL1)/RHO2)/SIAL1
      S13=              (W1/RHO - (VEC(3)*COAL1)/RHO2)/SIAL1
```

```
   82 IF(ABS(SIAL2) .GE. .001) GO TO 83
      S21= (1000.0)*(U2/RHO - VEC(1)/RHO2)
      S22= (1000.0)*(V2/RHO - VEC(2)/RHO2)
      S23= (1000.0)*(W2/RHO - VEC(3)/RHO2)
      GO TO 84
   83 S21=              (U2/RHO - (VEC(1)*COAL2)/RHO2)/SIAL2
      S22=              (V2/RHO - (VEC(2)*COAL2)/RHO2)/SIAL2
      S23=              (W2/RHO - (VEC(3)*COAL2)/RHO2)/SIAL2
   84 IF(ABS(SIAL3) .GE. .001) GO TO 85
      S31= (1000.0)*(U3/RHO - VEC(1)/RHO2)
      S32= (1000.0)*(V3/RHO - VEC(2)/RHO2)
      S33= (1000.0)*(W3/RHO - VEC(3)/RHO2)
      GO TO 86
   85 S31 =             (U3/RHO - (VEC(1)*COAL3)/RHO2)/SIAL3
      S32 =             (V3/RHO - (VEC(2)*COAL3)/RHO2)/SIAL3
      S33 =             (W3/RHO - (VEC(3)*COAL3)/RHO2)/SIAL3
   86 H(1,1) = S11
      H(1,2) = S12
      H(1,3) = S13
      H(2,1) = S21
      H(2,2) = S22
      H(2,3) = S23
      H(3,1) = S31
      H(3,2) = S32
      H(3,3) = S33
      NR=3
      GO TO (200,200,200,88,89,200,200,90,91,92,93),IAUG
   88 H(1,9) =1.0
      H(2,10) = 1.0
      H(3,11) = 1.0
      GO TO 200
89    H(1,7)=(-S11*DELXA-S12*DELYA-S13*DELZA)/DELAXS
      H(1,8)=(-S11*DELXE-S12*DELYE-S13*DELZE)/DELECC
      H(1,9)=(-S11*DELXI-S12*DELYI-S13*DELZI)/DELICL
      H(2,7)=(-S21*DELXA-S22*DELYA-S23*DELZA)/DELAXS
      H(2,8)=(-S21*DELXE-S22*DELYE-S23*DELZE)/DELECC
      H(2,9)=(-S21*DELXI-S22*DELYI-S23*DELZI)/DELICL
      H(3,7)=(-S31*DELXA-S32*DELYA-S33*DELZA)/DELAXS
      H(3,8)=(-S31*DELXE-S32*DELYE-S33*DELZE)/DELECC
      H(3,9)=(-S31*DELXI-S32*DELYI-S33*DELZI)/DELICL
      GO TO 200
   90 H(1,12) = 1.0
      H(2,13) = 1.0
      H(3,14) = 1.0
      GO TO 200
91    H(1, 9)=(-S11*DELXA-S12*DELYA-S13*DELZA)/DELAXS
      H(1,10)=(-S11*DELXE-S12*DELYE-S13*DELZE)/DELECC
      H(1,11)=(-S11*DELXI-S12*DELYI-S13*DELZI)/DELICL
      H(2,9)=(-S21*DELXA-S22*DELYA-S23*DELZA)/DELAXS
      H(2,10)=(-S21*DELXE-S22*DELYE-S23*DELZE)/DELECC
      H(2,11)=(-S21*DELXI-S22*DELYI-S23*DELZI)/DELICL
      H(3,9)=(-S31*DELXA-S32*DELYA-S33*DELZA)/DELAXS
      H(3,10)=(-S31*DELXE-S32*DELYE-S33*DELZE)/DELECC
      H(3,11)=(-S31*DELXI-S32*DELYI-S33*DELZI)/DELICL
      GO TO 200
   92 H(1,9) = 1.0
      H(2,10)= 1.0
      H(3,11)= 1.0
      H(1,13)=(-S11*DELXA-S12*DELYA-S13*DELZA)/DELAXS
      H(1,14)=(-S11*DELXE-S12*DELYE-S13*DELZE)/DELECC
```

```
      H(1,15)=(-S11*DELXI-S12*DELYI-S13*DELZI)/DELICL
      H(2,13)=(-S21*DELXA-S22*DELYA-S23*DELZA)/DELAXS
      H(2,14)=(-S21*DELXE-S22*DELYE-S23*DELZE)/DELECC
      H(2,15)=(-S21*DELXI-S22*DELYI-S23*DELZI)/DELICL
      H(3,13)=(-S31*DELXA-S32*DELYA-S33*DELZA)/DELAXS
      H(3,14)=(-S31*DELXE-S32*DELYE-S33*DELZE)/DELECC
      H(3,15)=(-S31*DELXI-S32*DELYI-S33*DELZI)/DELICL
      GO TO 200
   93 H(1,14) = 1.0
      H(2,15) = 1.0
      H(3,16) = 1.0
      GO TO 200
    7 DO 101 I = 1,6
      VEC(I) = HECP(I) - HECV(I)
  101 CONTINUE
      RH = SQRT(VEC(1)*VEC(1) + VEC(2)*VEC(2) + VEC(3)*VEC(3))
      RH2=RH*RH
      RADNTP=RADIUS(NTP)*ALNGTH
      IF(IOBS.EQ.0) GO TO 102
      VECTOR(1)=2.*ASIN(RADNTP/RH)
      GO TO 200
102   DENOM = RH2*  SQRT(RH2   - RADNTP*RADNTP)
      AD1 = (2.0 * RADNTP     *VEC(1))/DENOM
      AD2 = (2.0 * RADNTP     *VEC(2))/DENOM
      AD3 = (2.0 * RADNTP     *VEC(3))/DENOM
      H(1,1) = AD1
      H(1,2) = AD2
      H(1,3) = AD3
      NR=1
      GO TO(200,200,200,103,104,200,200,105,106,107,108),IAUG
  103 H(1,12) = 1.0
      GO TO 200
104   H(1,7)=(-AD1*DELXA-AD2*DELYA-AD3*DELZA)/DELAXS
      H(1,8)=(-AD1*DELXE-AD2*DELYE-AD3*DELZE)/DELECC
      H(1,9)=(-AD1*DELXI-AD2*DELYI-AD3*DELZI)/DELICL
      GO TO 200
  105 H(1,15) = 1.0
      GO TO 200
106   H(1,9)=(-AD1*DELXA-AD2*DELYA-AD3*DELZA)/DELAXS
      H(1,10)=(-AD1*DELXE-AD2*DELYE-AD3*DELZE)/DELECC
      H(1,11)=(-AD1*DELXI-AD2*DELYI-AD3*DELZI)/DELICL
      GO TO 200
  107 H(1,12) = 1.0
      H(1,13)=(-AD1*DELXA-AD2*DELYA-AD3*DELZA)/DELAXS
      H(1,14)=(-AD1*DELXE-AD2*DELYE-AD3*DELZE)/DELECC
      H(1,15)=(-AD1*DELXI-AD2*DELYI-AD3*DELZI)/DELICL
      GO TO 200
  108 H(1,17) = 1.0
200   RETURN
      END
```

```
      SUBROUTINE TRANS(ICODE,X,Y,Z,VX,VY,VZ,XE,YE,ZE,VXE,VYE,VZE,EPS,
     $ICODE2)
C
C     IF THE POSITION OF THE SPACECRAFT IS IN GEOCENTRIC EQUATORIAL
C     COORDINATES THIS SUBROUTINE CONVERTS TO
C          1)  GEOCENTRIC ECLIPTIC RECTANGULAR COORDINATES
C          2)  HELIOCENTRIC ECLIPTIC RECTANGULAR COORDINATES
C     THE VARIABLE ICODE DETERMINES WHICH OF THE ABOVE OPTIONS WILL BE
C     EXERCISED BY
C          ICODE = 1 -- OPTION 1)
C          ICODE = 2 -- OPTION 2)
C     SUBROUTINE ARGUMENTS ARE DISCUSSED IN THE FOLLOWING TABLE
C          X    -- X COMPONENT  OF THE SPACECRAFT IN GEO-EQUATORIAL
C          Y    -- Y COMPONENT  OF THE SPACECRAFT IN GEO-EQUATORIAL
C          Z    -- Z COMPONENT  OF THE SPACECRAFT IN GEO-EQUATORIAL
C          VX   -- X VELOCITY COMPONENT OF THE SPACECRAFT IN GEO-EQUAT.
C          VY   -- Y VELOCITY COMPONENT OF THE SPACECRAFT IN GEO-EQUAT.
C          VZ   -- Z VELOCITY COMPONENT OF THE SPACECRAFT IN GEO-EQUAT.
C          XE   -- X COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          YE   -- Y COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          ZE   -- Z COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          VXE -- X VELOCITY COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          VYE -- Y VELOCITY COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          VZE -- Z VELOCITY COMPONENT OF THE EARTH IN HELIO-ECLIPTIC
C          EPS -- OBLIQUITY OF THE EARTH
C
C     NOTE -- IF THE POSITION OF THE SPACECRAFT IS IN GEOCENTRIC
C     ECLIPTIC COORDINATES AND THE USER WISHES TO RETURN HELIOCENTRIC
C     ECLIPTIC COORDINATES, SET THE ARGUMENT
C          ICODE2 = 2
C     ANY OTHER VALUE FOR THIS VARIABLE WILL ALLOW THE PROGRAM TO ASSUME
C     THE COORDINATES ARE IN GEOCENTRIC EQUATIORIAL
C
C     THE NEW COORDINATES OF THE SPACECRAFT ARE THEN RETURNED IN THE
C     LOCATIONS X,Y,Z,VX,VY,VZ
C
C
C     NOTE -- THE HELIOCENTRIC ECLIPTIC COORDINATES OF THE EARTH ARE
C     NECESSARY ONLY IF OPTION 2 IS SPECIFIED.  IF OPTION 1 IS
C     INDICATED ZEROS SHOULD BE USED IN THE CALL STATEMENT FOR
C     THE LAST SIX VARIABLES IN THE ARGUMENT LIST AS THEY WILL BE
C     IGNORED -- HOWEVER, IT IS NECESSARY TO FILL THESE LOCATIONS SO
C     THAT THE PROPER NUMBER OF ARGUMENTS WILL APPEAR.
C
      IF (ICODE2-2) 1,10,1
1     CE=COS(EPS)
      SE=SIN(EPS)
      DUM=Y*CE+Z*SE
      Z=Z*CE-Y*SE
      Y=DUM
      DUM=VY*CE+VZ*SE
      VZ=VZ*CE-VY*SE
      VY=DUM
      IF(ICODE-1) 20,20,10
10    X=X+XE
      Y=Y+YE
      Z=Z+ZE
      VX=VX+VXE
      VY=VY+VYE
      VZ=VZ+VZE
```

```
20      RETURN
        END
```

```
      SUBROUTINE VARADA (RI,XSIP,XSIV,TEVN,TSI,ADA,BS,BDTS,BDRS)
C
C
C     THIS SUBROUTINE CALCULATES THE ADA MATRIX WHICH ARISES FROM
C     VARIATIONS IN B DOT T, B DOT R, AND THE TIME AT WHICH THE SPHERE
C     OF INFLUENCE IS ENCOUNTERED FROM VARIATIONS IN THE INITIAL
C     CONDITIONS OF THE STATE VECTOR.
C
C     VARADA USES THE FOLLOWING SUBROUTINES
C        NTM
C        BLOCK DATA
C
C
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON/TIM /DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION ADA(3,6),XC(6),RI(6),XSIP(3),XSIV(3),RF(6),RSI1(3),
     $VSI1(3)
      B1=BS
      BDT1=BDTS
      BDR1=BDRS
      TSI1=TSI
      DSI1=DSI
      ISP=ISP2
      ISP2=NTP
      TRTM1=TEVN
      IPR=IPRINT
      IPRINT=1
      N=1
5     DO 10 I=1,6
10    XC(I)=RI(I)
      ISPH=0
15    IF(N-4) 20,30,30
20    XC(N)=XC(N)+FACP
      GO TO 40
30    XC(N)=XC(N)+FACV
40    DELTM=FNTM-TEVN
      CALL NTM(XC,RF,NTMC,-1)
      IF(ISPH.NE.0) GO TO 50
      WRITE(6,1000)
1000  FORMAT(///8X*VEHICLE DID NOT REACH SPHERE OF INFLUENCE IN NUMERICA
     $L DIFFERENCING TO DETERMINE BDT, BDR, TSI VARIATIONS*/8X*RETURNING
     $ TO BASIC CYCLE*///)
      GO TO 100
50    TSI =DSI-DATEJ
      IF(N-4)60,70,70
60    ADA(1,N)=(BDT-BDT1)/FACP
      ADA(2,N)=(BDR-BDR1)/FACP
      ADA(3,N)=(TSI-TSI1)/FACP
      GO TO 80
70    ADA(1,N)=(BDT-BDT1)/FACV
      ADA(2,N)=(BDR-BDR1)/FACV
      ADA(3,N)=(TSI-TSI1)/FACV
80    N=N+1
      ISPH=0
```

```
      IF(N-6) 5, 5,90
90    DO 91 I=1,3
      RSI(I)=XSIP(I)
91    VSI(I)=XSIV(I)
      B=B1
      BDT=BDT1
      BDR=BDR1
      DSI=DSI1
      TSI=TSI1
      ISP2=ISP
      IPRINT=IPR
      ISPH=1
100   RETURN
      END
```

```
      SUBROUTINE VARSIM(RI1,TEVN,TSI,ADA)
      COMMON/CONST/OMEGA,EPS,NST,SAL(3),SLAT(3),SLON(3),DNCN(3),MNCN(12)
      COMMON /CONST2/U1,U2,U3,V1,V2,V3,W1,W2,W3,FOP,FOV
      COMMON/EVENT/NEV,TEV(50),IEVNT(50),IHYP1,IEIG,TPT2(20),
     $ICDT3(20),NPE,NGE,IPOL,IIPOL,ICDQ3(20),SIGRES,SIGPRO,SIGALP,SIGBET
     $,NEV1,NEV2,NEV3,NEV4,NQE
      COMMON/GUI/PG(17,17),XG(6),TG,EM(2,6)
      COMMON/MISC/ACC,IDNF,ICOOR,ITR,IMNF,FACP,FACV,ISP2,BIA(12),IPGN
      COMMON /NAME/MDNM(4,2),EVNM(4),MNNAME(12,3),CMPNM(11,17)
      COMMON /SIM1/XI1(17),XF1(17),ADEVX(17),EDEVX(17),W(17),Z(17),
     $ANOIS(17),RES(4),EY(4),AY(4),AR(4,4),ZI(17),ADEVXB(17)
      COMMON/SIM2/NB1(11),ACC1,NBOD1
      COMMON/STM/P(17,17),PSI(17,17),Q(17,17),H(4,17),R(4,4),AK(17,4)
     $,PB(17,17),PSIP(17,17),HPHR(4,4)
      COMMON/STVEC/XI(17),XF(17),NDIM,IAUG,XB(17)
      COMMON/TIM/DATEJ,TRTM1,DELTM,FNTM,UNIVT,TRTMB
      COMMON/TRAJCD/NTMC,ISTMC,ISTM1,DTMAX,NDACC,ACCND
      COMMON/TRJ/ISOI1,ISOI2,ISOI3,ICA1,ICA2,ICA3,RCA1(6),RCA2(6),
     $RCA3(6),RSOI1(3),RSOI2(3),RSOI3(3),VSOI1(3),VSOI2(3),VSOI3(3),
     $TCA1,TCA2,TCA3,TSOI1,TSOI2,TSOI3,BSI1,BSI2,BSI3,BDTSI1,BDTSI2,
     $BDTSI3,BDRSI1,BDRSI2,BDRSI3
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RI1(6),ADA(3,6),RSIS(3),VSIS(3),XC(6),RF1(6)
      BS=B
      BDTS=BDT
      BDRS=BDR
      DO 10 I=1,3
      RSIS(I)=RSI(I)
10    VSIS(I)=VSI(I)
      IPR=IPRINT
      ISPS=ISP2
      IPRINT=1
      ISP2=NTP
      TRTM1=TEVN
      N=1
20    DO 30 I=1,6
30    XC(I)=RI1(I)
      ISPH=0
      IF(N-4) 40,50,50
40    XC(N)=XC(N)+FACP
      GO TO 60
50    XC(N)=XC(N)+FACV
60    CALL NTM(XC,RF1,NTMC,-2)
      IF(ISPH.EQ.1) GO TO 70
      WRITE(6,1000)
1000  FORMAT(///1X,130(1H*)//8X*NOTE--*//8X*VEHICLE DID NOT REACH SPHERE
     $OF INFLUENCE IN NUMERICAL DIFFERENCING TO DETERMINE BDT,BDR,TSI,
     $VARIATIONS*/8X*RETURNING TO BASIC CYCLE*///1X,130(1H*))
      GO TO 120
70    TSI1=DSI-DATEJ
      IF(N-4) 80,90,90
80    ADA(1,N)=(BDT-BDTS)/FACP
      ADA(2,N)=(BDR-BDRS)/FACP
      ADA(3,N)=(TSI1-TSI)/FACP
      GO TO 100
90    ADA(1,N)=(BDT-BDTS)/FACV
      ADA(2,N)=(BDR-BDRS)/FACV
      ADA(3,N)=(TSI1-TSI)/FACV
```

```
100   N=N+1
      IF(N.LE.6) GO TO 20
      DO 110 I=1,3
      RSI(I)=RSIS(I)
110   VSI(I)=VSIS(I)
      B=BS
      BDT=BDTS
      BDR=BDRS
      DSI=TSI+DATEJ
      IPRINT=IPR
      ISP2=ISPS
120   RETURN
      END
```

```
      SUBROUTINE VECTOR
C
C
C
C     THIS SUBROUTINE CALCULATES THE VECTOR ORBITAL ELEMENTS K, E,
C     COMPUTES THE SPACECRAFT FINAL POSITION ON THE ORBIT TO ACCURATELY
C     APPROXIMATE THE DESIRED TIME INTERVAL AND THEN COMPUTES THE CONIC
C     SECTION TIME OF FLIGHT.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
C VECTOR ORBITAL ELEMENTS
C     K = RVSB X VVSB
400   V(16,2)=V(9,3)*V(11,4)-V(9,4)*V(11,3)
      V(16,3)=V(9,4)*V(11,2)-V(9,2)*V(11,4)
      V(16,4)=V(9,2)*V(11,3)-V(9,3)*V(11,2)
C     E = RVSB/ABS(RVSB) - (K X RVSB)/MUV
403   V(14,2)=-V(9,2)/V(9,1)-(V(16,3)*V(11,4)-V(16,4)*V(11,3))/V(7,7)
      V(14,3)=-V(9,3)/V(9,1)-(V(16,4)*V(11,2)-V(16,2)*V(11,4))/V(7,7)
      V(14,4)=-V(9,4)/V(9,1)-(V(16,2)*V(11,3)-V(16,3)*V(11,2))/V(7,7)
404   V(14,5)=V(14,2)*V(14,2)+V(14,3)*V(14,3)+V(14,4)*V(14,4)
      V(14,1)=SQRT(V(14,5))
      V(16,5)=V(16,2)*V(16,2)+V(16,3)*V(16,3)+V(16,4)*V(16,4)
      V(16,1)=SQRT(V(16,5))
      V(13,5)=1.-V(14,5)
      V(13,6)= SQRT( ABS(V(13,5)))
      V(13,7)=V(16,5)/V(7,7)
      IF(ITRAT.EQ.3) RETURN
C SPACECRAFT FINAL POSITION AND VELOCITY
C     DELTA TAU = DELTA T + (DELTA T)**2 * KAPPA
      V(6,7)=V(7,6)+V(7,6)*V(7,6)*V(6,7)
C     SIGMA = RVSB + DELTA TAU  DOT  VVSB
410   V(9,5)=V(9,2)+V(6,7)*V(11,2)
      V(9,6)=V(9,3)+V(6,7)*V(11,3)
      V(9,7)=V(9,4)+V(6,7)*V(11,4)
C     B = K**2/MUV*(E DOT SIGMA + ABS(SIGMA))
      V(8,7)=V(13,7)/(V(14,2)*V(9,5)+V(14,3)*V(9,6)+V(14,4)*V(9,7)+ SQRT
     1(V(9,5)*V(9,5)+V(9,6)*V(9,6)+V(9,7)*V(9,7)))
411   V(10,2)=V(8,7)*V(9,5)
      V(10,3)=V(8,7)*V(9,6)
      V(10,4)=V(8,7)*V(9,7)
      V(10,1) = SQRT(V(10,2)*V(10,2)+V(10,3)*V(10,3)+V(10,4)*V(10,4))
414   V(12,5)=V(14,2)+V(10,2)/V(10,1)
      V(12,6)=V(14,3)+V(10,3)/V(10,1)
      V(12,7)=V(14,4)+V(10,4)/V(10,1)
413   V(12,2)=(V(16,3)*V(12,7)-V(16,4)*V(12,6))/V(13,7)
      V(12,3)=(V(16,4)*V(12,5)-V(16,2)*V(12,7))/V(13,7)
      V(12,4)=(V(16,2)*V(12,6)-V(16,3)*V(12,5))/V(13,7)
C     VVSE = (K X (E + RVSE/ABS(RVSE)))/(K**2/MUV)
C     RSE=RVSE + RVE
      V(2,2)=V(10,2)+V(6,2)
      V(2,3)=V(10,3)+V(6,3)
      V(2,4)=V(10,4)+V(6,4)
```

```
C      VSE = VVSE + VVE
       V(4,2)=V(12,2)+V(10,5)
       V(4,3)=V(12,3)+V(10,6)
       V(4,4)=V(12,4)+V(10,7)
C KEPLERIAN TIME OF FLIGHT
       IF(V(14,1)) 520,510,520
510    V(12,1)=V(9,1)*V(9,1)
       V(13,1)=V(10,2)*V(9,2)+V(10,3)*V(9,3)+V(10,4)*V(9,4)
       DUM=V(16,1)*V(9,1)
       V(11,5)=(V(16,3)*V(9,4)-V(16,4)*V(9,3))/DUM
       V(11,6)=(V(16,4)*V(9,2)-V(16,2)*V(9,4))/DUM
       V(11,7)=(V(16,2)*V(9,3)-V(16,3)*V(9,2))/DUM
       GO TO 530
520    V(12,1)=V(13,7)/V(13,5)-V(9,1)
       V(13,1)=V(13,7)/V(13,5)-V(10,1)
       DUM=V(16,1)*V(14,1)
       V(11,5)=(V(16,3)*V(14,4)-V(16,4)*V(14,3))/DUM
       V(11,6)=(V(16,4)*V(14,2)-V(16,2)*V(14,4))/DUM
       V(11,7)=(V(16,2)*V(14,3)-V(16,3)*V(14,2))/DUM
530    IF(V(13,5)) 550,540,560
540    V(8,7)=.5*V(16,1)
       V(15,5)=2./(V(9,1)-V(13,7))
       V(16,5)=2./(V(10,1)-V(13,7))
       V(15,5)=(V(11,5)*V(9,2)+V(11,6)*V(9,3)+V(11,7)*V(9,4))/V(15,5)
       V(16,5)=(V(11,5)*V(10,2)+V(11,6)*V(10,3)+V(11,7)*V(10,4))/V(16,5)
       DUM=V(13,7)*V(13,7)*V(13,7)/3.
       V(15,6)=DUM*V(15,5)*V(15,5)*V(15,5)
       V(16,6)=DUM*V(16,5)*V(16,5)*V(16,5)
       GO TO 660
550    V(15,5)=V(13,7)/V(13,6)
       V(8,7)=V(13,5)*V(7,7)/(V(16,1)*V(15,5))
       V(16,5)=(V(11,5)*V(10,2)+V(11,6)*V(10,3)+V(11,7)*V(10,4))/V(15,5)
       V(15,5)=(V(11,5)*V(9,2)+V(11,6)*V(9,3)+V(11,7)*V(9,4))/V(15,5)
       V(15,6)=ALOG(V(15,5)+SQRT(V(15,5)*V(15,5)+1.))
       V(16,6)=ALOG(V(16,5)+SQRT(V(16,5)*V(16,5)+1.))
       GO TO 660
560    V(15,5)=V(13,7)/V(13,6)
       V(8,7)=V(13,5)*V(7,7)/(V(16,1)*V(15,5))
       V(16,5)=(V(11,5)*V(10,2)+V(11,6)*V(10,3)+V(11,7)*V(10,4))/V(15,5)
       V(15,5)=(V(11,5)*V(9,2)+V(11,6)*V(9,3)+V(11,7)*V(9,4))/V(15,5)
       DO 610 I=15,16
       IF(V(I,5)-1.0001) 580,580,570
570    CALL SPACE(2)
       WRITE(6,1000)
1000   FORMAT(/27H UNACCEPTABLE ERROR IN ATAN)
       KOUNT=-1
       RETURN
580    IF(V(I,5)-1.) 600,600,590
590    CALL SPACE(2)
       WRITE(6,1001)
1001   FORMAT(/25H ACCEPTABLE ERROR IN ATAN)
       V(I,6)=SIGN(.5*PI,V(I,5))
       GO TO 610
600    V(I,6)=ATAN(V(I,5)/SQRT(1.-V(I,5)*V(I,5)))
610    CONTINUE
       DO 650 I=12,13
       IF(V(I,1)) 620,650,650
620    IP3=I+3
       IF(V(IP3,5)) 630,640,640
630    V(IP3,6)=-PI-V(IP3,6)
```

```
      GO TO 650
640   V(IP3,6)=PI-V(IP3,6)
650   CONTINUE
660   V(7,5)=V(16,6)-V(15,6)+V(14,1)*(V(15,5)-V(16,5))
      IF(V(7,5))670,700,690
670   IF(V(8,7)) 690,680,680
680   V(7,5)=V(7,5)+2.*PI
690   V(7,5)=V(7,5)/V(8,7)
      V(6,7)=(V(6,7)-V(7,5))/(V(7,5)*V(7,5))
700   RETURN
      END
```

```
      SUBROUTINE VMASS
C
C
C
C     THIS SUBROUTINE DETERMINES THE VIRTUAL MASS DATA NEEDED IN THE
C     TRAJECTORY ANALYSIS.  THE V. M. POSITION, MAGNITUDE, VELOCITY, AND
C     MAGNITUDE RATE ARE CALCULATED USING FORMULAS (II-3) IN
C
C          NOVAK, D. H. -VIRTUAL MASS TECHNIQUE FOR COMPUTING SPACE
C          TRAJECTORIES-, FINAL REPORT, CONTRACT NO. NAS 9-4370,
C          ER 14045, MARTIN, BALTIMORE DIVISION, JANUARY, 1966.  PG. 8.
C
C     GIVEN THE POSITIONS OF THE SPACECRAFT AND THE PLANETS.
C
C
C
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
C VIRTUAL MASS POSITION AND MAGNITUDE
      V(12,5)=0.
      DO 201 I=1,NBODY,4
      IP2=I+2
      DO 200 J=1,3
  200 F(IP2,J)=V(2,J+1)-F(I,J)
      F(IP2,4)=SQRT(F(IP2,1)*F(IP2,1)+F(IP2,2)*F(IP2,2)+F(IP2,3)*
     1F(IP2,3))
      F(I+3,4)=F(I,4)/(F(IP2,4)*F(IP2,4)*F(IP2,4))
  201 V(12,5)=V(12,5)+F(I+3,4)
      DO 203 J=1,3
      JP1=J+1
      V(6,JP1)=0.
      DO 202 I=1,NBODY,4
  202 V(6,JP1)=V(6,JP1)+F(I+3,4)*F(I,J)
      V(6,JP1)=V(6,JP1)/V(12,5)
  203 V(10,JP1)=V(2,JP1)-V(6,JP1)
      V(10,1)=SQRT(V(10,2)*V(10,2)+V(10,3)*V(10,3)+V(10,4)*V(10,4))
      V(6,1)=V(10,1)*V(10,1)*V(10,1)*V(12,5)
C VIRTUAL MASS VELOCITY AND MAGNITUDE RATE
      V(12,6)=0.
      DO 301 I=1,NBODY,4
      IP1=I+1
      IP2=I+2
      IP3=I+3
      DO 300 J=1,3
  300 F(IP3,J)=V(4,J+1)-F(IP1,J)
      F(IP1,4)=3.*(F(IP2,1)*F(IP3,1)+F(IP2,2)*F(IP3,2)+F(IP2,3)*F(IP3,3)
     1)/(F(IP2,4)*F(IP2,4))
  301 V(12,6)=V(12,6)-F(IP1,4)*F(IP3,4)
      DO 303 J=1,3
      JP1=J+1
      V(8,JP1)=0.
      DO 302 I=1,NBODY,4
  302 V(8,JP1)=V(8,JP1)+F(I+3,4)*(F(I+1,J)-F(I,J)*F(I+1,4))
      V(8,JP1)=(V(8,JP1)-V(6,JP1)*V(12,6))/V(12,5)
  303 V(12,JP1)=V(4,JP1)-V(8,JP1)
```

```
 V(8,1)=V(6,1)*(3.*(V(10,2)*V(12,2)+V(10,3)*V(12,3)+V(10,4)*V(12,4)
1)/(V(10,1)*V(10,1))+V(12,6)/V(12,5))
 RETURN
 END
```

```
      SUBROUTINE VMP(RS,ACC,D1,TRTM,DELTM,RSF,ISP2)
C
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR GENERATING A VIRTUAL MASS
C     TRAJECTORY.
C
C     INPUT ARGUMENTS
C        RS       --  INITIAL POSITION AND VELOCITY OF VEHICLE
C        ACC      --  ACCURACY FIGURE WHICH DETERMINES TRUE ANOMALY
C                     INCREMENT
C        D1       --  JULIAN DATE OF INITIAL TRAJECTORY TIME
C        TRTM     --  INITIAL TRAJECTORY TIME
C        DELTM    --  NUMBER OF DAYS THE INTEGRATION IS TO CONTINUE
C                     UNLESS ISP2 IS NOT ZERO
C        ISP2     --  INTEGRATION CODE
C                     =0       THE INTEGRATION IS TO CONTINUE UNTIL A
C                              STOPPING CONDITION OCCURS
C                     .GT.0    THE INTEGRATION WILL STOP UPON ENCOUNTERING
C                              THE SPHERE OF INFLUENCE OF THE PLANET
C                              SPECIFIED BY ISP2
C
C     OUTPUT ARGUMENT
C        RSF      --  FINAL POSITION AND VELOCITY OF VEHICLE
C
C     COMMON ELEMENTS
C
C     INPUT
C        NBOD     --  NUMBER OF BODIES TO BE CONSIDERED IN ANALYSIS
C        NB       --  ARRAY OF CODES OF BODIES TO BE CONSIDERED
C        NTP      --  CODE NUMBER OF TARGET PLANET
C        ALNGTH   --  LENGTH UNITS PER A.U.
C        TM       --  TIME UNITS PER DAY
C        DELTP    --  PRINT INCREMENT (IN DAYS)
C        INPR     --  PRINT INCREMENT (INCREMENTS)
C        IPROB    --  INTEGER PROBLEM IDENTIFICATION
C
C     OUTPUT
C
C        RC       --  POSITION AND VELOCITY OF VEHICLE AT CLOSEST
C                     APPROACH TO TARGET PLANET
C        DC       --  TIME OF CLOSEST APPROACH
C        RSI      --  POSITION OF VEHICLE AT SPHERE OF INFLUENCE OF
C                     TARGET PLANET
C        VSI      --  VELOCITY OF VEHICLE AT SPHERE OF INFLUENCE OF
C                     TARGET PLANET
C        DSI      --  DATE AT SPHERE OF INFLUENCE OF TARGET PLANET
C        ISPH     --  SPHERE OF INFLUENCE CODE
C                     =0  VEHICLE DID NOT REACH SPHERE OF INFLUENCE
C                     =1  VEHICLE DID REACH SPHERE OF INFLUENCE
C        RVS      --  POSITION AND VELOCITY OF VEHICLE RELATIVE TO
C                     VIRTUAL MASS AT INITIAL TIME
C        VMU      --  MAGNITUDE OF VIRTUAL MASS
C        RE       --  POSITION AND VELOCITY OF EARTH AT FINAL TIME
C        RTP      --  POSITION AND VELOCITY OF TARGET PLANET AT FINAL
C                     TIME
C        B        --  B (CALCULATED AT SPHERE OF INFLUENCE)
C        BDT      --  B DOT T (CALCULATED AT SPHERE OF INFLUENCE)
C        BDR      --  B DOT R (CALCULATED AT SPHERE OF INFLUENCE)
C        RV       --  POSITION AND VELOCITY OF VIRTUAL MASS AT INITIAL
C                     TIME
```

```
C        DELTH    --  INCREMENTS OF TRUE ANOMALY USED IN RUN (DETERMINED
C                     BY ACC)
C        TIMINT   --  TOTAL CP TIME USED IN TRAJECTORY OF FINAL TIME
C        INCMNT   --  TOTAL INCREMENTS USED IN TRAJECTORY TO FINAL TIME
C                     (SHOULD BE INITIALIZED TO ZERO OUTSIDE OF VMP)
C
C
      COMMON/VM/NBOD,NB(11),NTP,ALNGTH,TM,DELTP,INPR,IPROB,RC(6),DC,
     $RSI(3),VSI(3),DSI,ISPH,RVS(6),VMU,B,BDT,BDR,DELTH,TIMINT,INCMT,
     $IEPHEM,ICL,IPRINT,RE(6),RTP(6),ICL2
      DIMENSION RS(6),RSF(6)
      COMMON /COM/V(16,7),F(44,4),PI,RAD
      COMMON /COM/ITRAT,KOUNT,INCMNT,INCPR,INC,IPR
      COMMON/COM/NBODYI,NBODY,IPRT(4)
      COMMON/COM/KL,IPG,LINCT,LINPGE
      COMMON/BLK/T,PMASS(11),CN(80),ST(50),EMN(15),SMJR(18)
      COMMON /BLK/RADIUS(11),RMASS(11),NO(11),ELMNT(80),SPHERE(11),XP(6)
      COMMON /PRT/MONTH(12),PLANET(11)
      CALL CPWMS(TIM1)
      KL=IPROB
      DO 16J=1,7
      DO 16I=1,16
16    V(I,J)=0.
      V(1,6)=ALNGTH
      V(1,5)=TM
      V(4,7)=V(1,6)/V(1,5)
      V(5,5)=V(1,6)*V(4,7)
      V(6,5)=V(5,5)*V(4,7)
      V(6,6)=V(6,5)/V(1,5)
      V(4,5)=ACC
      V(3,1)=D1
      V(1,1)=TRTM
      V(2,5)=V(1,1)+DELTM
      V(3,5)=DELTP
      NBODYI=NBOD
      INCPR=INPR
      DO 15 I=1,NBODYI
15    NO(I)=NB(I)
      INCMNT=INCMT
      CALL INPUTZ(RS,NTP,IPRINT)
2     IF(IEPHEM.EQ.0) GO TO 8
      DO 6 I=1,NBODYI
      J=NO(I)
6     CALL ORB(J,V(4,1))
8     CALL EPHEM(0,V(4,1),NBODYI)
    3 CALL VMASS
      IF(ITRAT.EQ.1)GO TO 4
      IF(ITRAT.EQ.2) GO TO 7
C INITIALIZATION OF VIRTUAL MASS-DEPENDENT VALUES
      V(7,7) = V(6,1)
      DO 600 J=2,4
      V(10,J+3)=V(8,J)
      V(9,J)=V(10,J)
  600 V(11,J)=V(12,J)
      V(9,1)=V(10,1)
      V(8,5)=1.
      V(5,1)=V(6,1)
      DO 17 I=1,NBODYI
      IF(NO(I)-NTP)17,18,17
17    CONTINUE
```

```
18    NTPI=4*I-1
      RCM1=F(NTPI,4)
      IF(ICL.NE.0) GO TO 700
      DC=V(3,1)
      DO 601 I=1,3
      RC(I)=F(NTPI,I)*V(1,6)
601   RC(I+3)=F(NTPI+1,1)*V(4,7)
700   ISPH1=0
      RVS(1)=V(10,2)*V(1,6)
      RVS(2)=V(10,3)*V(1,6)
      RVS(3)=V(10,4)*V(1,6)
      RVS(4)=V(12,2)*V(4,7)
      RVS(5)=V(12,3)*V(4,7)
      RVS(6)=V(12,4)*V(4,7)
      VMU=V(6,1)*V(6,5)
      DELTH=V(3,6)
      IF (IPRINT.NE.0) KOUNT=0
      GO TO 9
4     ITRAT=2
C VIRTUAL MASS AVERAGE MAGNITUDE AND VELOCITY
    5 V(7,7)=.5*V(5,1)+.5*V(6,1)
      DO 390 J=2,4
      V(10,J+3)=(V(6,J)-V(5,J))/V(7,6)
  390 V(11,J)=V(3,J)-V(10,J+3)
    9 CALL VECTOR
      IF(KOUNT .LT. 0) GO TO 996
      IF(ITRAT .EQ. 1) GO TO 2
      IF(ITRAT.EQ.2) GO TO 3
C VIRTUAL MASS AVERAGE ACCELERATIONS
    7 V(8,6)=(V(6,1)-V(5,1)-V(7,1)*V(7,6))/V(8,5)
      DO 340 J=2,4
  340 V(10,J+3)=(V(6,J)-V(5,J)-V(7,J)*V(7,6))/V(8,5)
      RCM2=F(NTPI,4)
      IF(ISPH-1) 389,395,389
389   IF(ISPH1.NE.0) GO TO 3890
      IF(1.025*SPHERE(NTP).LT.RCM2) GO TO 395
      V(4,6)=V(3,6)
      TP=ALOG(2.E-7)
      V(3,6)=EXP(1.13756474179255+.509713741462307*TP+.14560181279278E-2
     **TP*TP)
      ISPH1=1
3890  IF(SPHERE(NTP).LT.RCM2) GO TO 395
      ISPH=1
      DO 391 I=1,3
      RSI(I)=F(NTPI,I)*V(1,6)
391   VSI(I)=F(NTPI+1,I)*V(4,7)
      DSI=V(4,1)
      KOUNT=1
      TTG = PMASS(NTP)*V(6,5)
      CALL ACTB(RSI,VSI,TTG,B,BDT,BDR)
      IF(IPRINT.EQ.0) GO TO 392
      KOUNT=0
      GO TO 393
392   RCM=RCM2*V(1,6)
      VCM=SQRT(F(NTPI+1,1)*F(NTPI+1,1)+F(NTPI+1,2)*F(NTPI+1,2)+F(NTPI+1,
     $3)*F(NTPI+1,3))*V(4,7)
      D=DSI+2415020.
      WRITE(6,7000) PLANET(NTP),D  ,RSI,RCM,VSI,VCM,B,BDT,BDR
7000  FORMAT(1H1/////////*   SPACECRAFT PIERCED SPHERE OF INFLUENCE OF *
     $A10* AT DATE. . . .*F17.8//10X*POSITION. . . . .*4E20.11/
```

```
     $10X*VELOCITY. . . . .*4E20.11//10X*B . . . . *E20.11,5X,*B.T . . .
     $ . *E20.11,5X,*B.R . . . . *E20.11)
393   V(3,6)=V(4,6)
      JJ=0
3891   RCM=SQRT(RSI(1)*RSI(1)+RSI(2)*RSI(2)+RSI(3)*RSI(3))
      JJ=JJ+1
      DELR=SPHERE(NTP)*V(1,6)-RCM
      IF(DELR*DELR-4.*(V(1,6)/149598500.))3894,3894,3892
3892  DELT=RCM*DELR/(RSI(1)*VSI(1)+RSI(2)*VSI(2)+RSI(3)*VSI(3))
      DSI=DSI+DELT/V(1,5)
      DO 3893 I=1,3
3893  RSI(I)=RSI(I)+VSI(I)*DELT
      IF(JJ-10) 3891,3891,3894
3894   RCM=SQRT(RSI(1)*RSI(1)+RSI(2)*RSI(2)+RSI(3)*RSI(3))
      CALL ACTB(RSI,VSI,TTG,B,BDT,BDR)
      IF(IPRINT.NE.0) GO TO 394
      D=DSI+2415020.
      WRITE(6,7002) PLANET(NTP),D  ,RSI,RCM,VSI,VCM,B,BDT,BDR
7002  FORMAT(////////////*   INTERPOLATED INFORMATION AT SPHERE OF INFLU
     $ENCE*//
     $                    *   SPACECRAFT PIERCED SPHERE OF INFLUENCE OF *
     $A10* AT DATE. . . .*F17.8//10X*POSITION. . . . .*4E20.11/
     $10X*VELOCITY. . . . .*4E20.11//10X*B . . . . *E20.11,5X,*B.T . . .
     $ . *E20.11,5X,*B.R . . . . *E20.11)
394   IF(ISP2.NE.0) GO TO 995
395   IF(ICL-1) 396,400,396
396   IF (RCM2.LE.RCM1) GO TO 400
      DO 397 I=1,3
      RC(I)=F(NTPI,I)*V(1,6)
397   RC(I+3)=F(NTPI+1,I)*V(4,7)
      KOUNT=1
      ICL=1
      DC=V(4,1)
      IF(IPRINT.EQ.0) GO TO 398
      KOUNT=0
      GO TO 399
  398 RCM=RCM2*V(1,6)
      VCM=SQRT(F(NTPI+1,1)*F(NTPI+1,1)+F(NTPI+1,2)*F(NTPI+1,2)+F(NTPI+1,
     $3)*F(NTPI+1,3))*V(4,7)
      D=DC+2415020.
      WRITE(6,7001) PLANET(NTP),D ,(RC(I),I=1,3),RCM,(RC(I),I=4,6),VCM
7001  FORMAT(////////////*   SPACECRAFT REACHED POINT OF CLOSEST APPROAC
     $H OF *A10* AT DATE. . . . .*F17.8//10X*POSITION. . . . .*4E20.11/
     $10X*VELOCITY. . . . .*4E20.11)
C TEST FOR STOPPING CONDITIONS
  399 IF(ICL2.NE.0) GO TO 995
400   RCM1=RCM2
      IF(V(2,5).GT.V(2,1)+1.E-8) GO TO 401
      IF(IPRINT.NE.0) GO TO 995
      D=V(4,1)
      CALL TIME (D,IYR,IMO,IDAY,IHR,MIN,SEC,1)
      D=D+2415020.
      IMO = MONTH(IMO)
      CALL SPACE (4)
      WRITE (6,4000)IMO,IDAY,IHR,MIN,SEC,IYR,D
4000  FORMAT(//3X*CALENDAR DATE =*A10,I3*,*I3* HR,*I3* MIN,* F7.3* SEC,*
     $I5/3X*JULIAN DATE =*F17.8//53H STOPPING CONDITION--EXCEEDED MAXIMU
     $M TRAJECTORY TIME)
      GO TO 995
401   DO 403 J=1,NBODYI
```

```
      IP=NO(J)
      IF(F(4*J-1,4).GT.RADIUS(IP)) GO TO 403
      IF(IPRINT.NE.0) GO TO 995
      D=V(4,1)
      CALL TIME (D,IYR,IMO,IDAY,IHR,MIN,SEC,1)
      D=D+2415020.
      IMO = MONTH(IMO)
      CALL SPACE(5)
      WRITE (6,4010)IMO,IDAY,IHR,MIN,SEC,IYR,D ,PLANET(IP)
4010  FORMAT(//3X*CALENDAR DATE =*A10,I3*,*I3* HR,*I3* MIN,* F7.3* SEC,*
     $I5/3X*JULIAN DATE =*F17.8//  * STOPPING CONDITION--IMPACTED *
     $A10)
      GO TO 995
  403 CONTINUE
      IF(KOUNT .EQ. 0) GO TO 11
      KOUNT = 0
   10 CALL PRINT
      IF(KOUNT .LT. 0) GO TO 12
11    CALL ESTMT(D1,DELTM,TRTM)
      GO TO 5
  995 KOUNT=-1
996   RSF(1)=V(2,2)*V(1,6)
      RSF(2)=V(2,3)*V(1,6)
      RSF(3)=V(2,4)*V(1,6)
      RSF(4)=V(4,2)*V(4,7)
      RSF(5)=V(4,3)*V(4,7)
      RSF(6)=V(4,4)*V(4,7)
      DO 20 I=1,NBODYI
      IF(NO(I)-4) 20,30,20
20    CONTINUE
30    J=4*I-3
      DO 42 I=1,3
      RE(I)=F(J,I)*V(1,6)
42    RE(I+3)=F(J+1,I)*V(4,7)
      J=NTPI-2
      DO 45 I=1,3
      RTP(I)=F(J,I)*V(1,6)
45    RTP(I+3)=F(J+1,I)*V(4,7)
      IF(ICL.NE.0) GO TO 35
      DC=V(4,1)
      DO 32 I=1,3
      RC(I)=F(NTPI,I)*V(1,6)
32    RC(I+3)=F(NTPI+1,I)*V(4,7)
35    IF(IPRINT.EQ.0) GO TO 10
   12 CALL CPWMS(TIM2)
      TIMIN = TIM2 - TIM1
      IF(IPRINT.NE.0) GO TO 50
      WRITE(6,4011) TIMIN
 4011 FORMAT(///10X,*TOTAL CP TIME USED IN THIS INTEGRATION . . . *5X,F1
     $0.3*  SEC*)
50    TIMINT=TIMINT+TIMIN
      INCMT=INCMNT
      RETURN
      END
```

VIII. EXAMPLE RUNS

This chapter presents sample runs for the various operational modes of STEAP. The test cases as shown on the following pages are not complete runs. Each example run was terminated after a few seconds of computer time. In this manner, typical output from each mode of operation can be presented.

The first problem shows typical output from the trajectory mode of STEAP. The targeting mode printout is shown in problem 2. The third and fourth problems represent typical computer printout from the error analysis and simulation modes.

```
INPUT  DATA  FOR  PROBLEM.....   1

 MODE TO BE EXECUTED. . .TRAJECTORY MODE

  LAUNCH DATE              7 24  9 25 47.639 1973     JULIAN DATE . . . 2441887.89291248

  FINAL DATE               2 20  0  0  0.    1974     JULIAN DATE . . . 2442098.50000000

  INITIAL TRAJECTORY TIME =    0.

  AUGMENTATION CODE. . . . .  1

  INITIAL STATE VECTOR
    GEOCENTRIC ECLIPTIC COORDINATES
         -1.09005292E+03
         -6.55005114E+03
          7.56015372E+01
          9.40152732E+00
         -2.87640366E+00
          6.20079494E+00

  INITIAL STATE VECTOR
    HELIOCENTRIC ECLIPTIC COORDINATES
          7.90303912E+07
         -1.29799646E+08
          7.56015372E+01
          3.43585796E+01
          1.25084614E+01
          6.20079494E+00


  NOMINAL TRAJECTORY CODE. . . 2

  NOMINAL TRAJECTORY INFORMATION

    BODIES TO BE CONSIDERED
      SUN
      EARTH
      MARS
      JUPITER
      MOON

    TARGET PLANET. . .MARS

  UNITS
     1.49598500E+08/A.U.                   8.64000000E+04/DAY

  ORBITAL ELEMENTS WILL BE CALCULATED AT EVERY TIME INTERVAL

  OUTPUT FROM VIRTUAL MASS PROGRAM WILL BE PRINTED AS USUAL

  VIRTUAL MASS PROGRAM WILL INTEGRATE UNTIL REACHING A NORMAL STOPPING CONDITION

  ACCURACY FIGURE. . . . . 5.00000E-06

  PRINT INTERVALS
     3.00000E+02 DAYS
         10000 INCREMENTS
```

```
V I R T U A L   M A S S   P R O G R A M   F O R   C O M P U T I N G   S P A C E   T R A J E C T O R I E S
                                                                                  PROBLEM   1    PAGE   1

UNITS . . . . . . . .   1.49598500000E+08 LNGTH/A.U.        8.64000000000E+04 TIME/DAY
 5 BODIES . . . . . .    1    4    5    6   11
LAUNCH DATE . . . . .      JULY 24,  9 HR, 25 MIN, 47.638 SEC, 1973. . . .JULIAN DATE. . 2441887.89291248
ENCOUNTER DATE. . .   FEBRUARY 20,  0 HR,  0 MIN,  0.    SEC, 1974. . . .JULIAN DATE. . 2442098.50000000
INITIAL TIME. . . .     0.
ACCURACY. . . . . .     5.00000000000E-06          TRUE ANOMALY INCREMENT. . .   7.69558353817E-03
```

```
VIRTUAL MASS PROGRAM FOR COMPUTING SPACE TRAJECTORIES
                                                              PROBLEM   1     PAGE   2

                                    X - COMP.          Y - COMP.          Z - COMP.          RESULTANT

TRAJECTORY TIME =    0.                TOTAL TIME INCREMENTS =     0

SPACECRAFT INERTIAL TRAJECTORY
    POSITION . . . . . . . . . . . . .  7.90303912037E+07  -1.29799645676E+08   7.56015371597E+01   1.51966281627E+08
    VELOCITY . . . . . . . . . . . . .  3.43585796040E+01   1.25084614232E+01   6.20079493827E+00   3.70867018950E+01

******************************************************************************************************************

CALENDAR DATE =      JULY 24,  9 HR, 25 MIN, 47.638 SEC, 1973
JULIAN DATE   =  2441887.89291248

EPHEMERIS DATA
    POSITION OF SUN      . . . . . .   0.                  0.                  0.                  0.
    VELOCITY OF SUN      . . . . . .   0.                  0.                  0.                  0.

    POSITION OF EARTH    . . . . . .   7.90314812566E+07  -1.29793095625E+08   0.                  1.51961253948E+08
    VELOCITY OF EARTH    . . . . . .   2.49570522842E+01   1.53848650825E+01   0.                  2.93180581267E+01

    POSITION OF MARS     . . . . . .   1.85975665790E+08  -8.98917600738E+07  -6.44851359101E+06   2.06661704538E+08
    VELOCITY OF MARS     . . . . . .   1.14881882815E+01   2.38768947350E+01   2.20800870369E-01   2.64977984972E+01

    POSITION OF JUPITER  . . . . . .   4.53809654071E+08  -6.12428578547E+08  -7.70673178353E+06   7.62280368148E+08
    VELOCITY OF JUPITER  . . . . . .   1.03479467763E+01   8.39888242769E+00  -2.65760226686E-01   1.33301109154E+01

    POSITION OF MOON     . . . . . .   7.92539490227E+07  -1.29502609306E+08   2.53921455478E+04   1.51829229391E+08
    VELOCITY OF MOON     . . . . . .   2.40968907669E+01   1.60098326356E+01  -7.85903022096E-02   2.89306249861E+01

******************************************************************************************************************

SPACECRAFT RELATIVE TRAJECTORIES
    POSITION REL. TO SUN     . . .   7.90303912037E+07  -1.29799645676E+08   7.56015371597E+01   1.51966281627E+08
    VELOCITY REL. TO SUN     . . .   3.43585796040E+01   1.25084614232E+01   6.20079493827E+00   3.70867018950E+01

    POSITION REL. TO EARTH   . . .  -1.09005292338E+03  -6.55005113463E+03   7.56015371597E+01   6.64056479785E+03
    VELOCITY REL. TO EARTH   . . .   9.40152731987E+00  -2.87640365928E+00   6.20079493827E+00   1.16237804446E+01

    POSITION REL. TO MARS    . . .  -1.06945274586E+08  -3.99078856020E+07   6.44858919255E+06   1.14330728118E+08
    VELOCITY REL. TO MARS    . . .   2.28703913226E+01  -1.13684333118E+01   5.97999406790E+00   2.62308292714E+01

    POSITION REL. TO JUPITER . . .  -3.74779262868E+08   4.82628932871E+08   7.70680738507E+06   6.11105209927E+08
    VELOCITY REL. TO JUPITER . . .   2.40106328277E+01   4.10957899549E+00   6.46655516496E+00   2.52034811883E+01

    POSITION REL. TO MOON    . . .  -2.23557818968E+05  -2.97036369952E+05  -2.53165440107E+04   3.72625322404E+05
    VELOCITY REL. TO MOON    . . .   1.02616888372E+01  -3.50137121241E+00   6.27938524048E+00   1.25296662827E+01
```

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
VIRTUAL MASS DATA				
VIRTUAL MASS POSITION	7.90314790767E+07	-1.29793091999E+08	1.76278393982E-03	1.51961249717E+08
VIRTUAL MASS VELOCITY	2.49557084715E+01	1.53870999491E+01	1.13511913734E-06	2.93180870822E+01
SPACECRAFT POS. REL. TO V.M.	-1.08787295138E+03	-6.55367696888E+03	7.55997743756E+01	6.64378391406E+03
SPACECRAFT VEL. REL. TO V.M.	9.40287113253E+00	-2.87863852590E+00	6.20079380315E+00	1.16254199531E+01
KEPLER (ANG. MOM.) VECTOR	-4.04203751134E+04	7.45653079164E+03	6.47549729718E+04	7.66981948962E+04
ECCENTRICITY VECTOR	-4.19058225111E-01	-1.16677959912E+00	-1.27223639181E-01	1.24626244159E+00
V.M. MAGN. = 3.99179976532E+05				
V.M. MAGN. RATE = 9.58028455382E-01				

		X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
V.M. RELATIVE POSITIONS					
POSITION REL. TO SUN	. . .	7.90314790767E+07	-1.29793091999E+08	1.76278393982E-03	1.51961249717E+08
POSITION REL. TO EARTH	. . .	-2.17997217178E+00	3.62583446503E+00	1.76278393982E-03	4.23071593763E+00
POSITION REL. TO MARS	. . .	-1.06944186713E+08	-3.99013319250E+07	6.44851359277E+06	1.14327418797E+08
POSITION REL. TO JUPITER	. . .	-3.74778174995E+08	4.82635466548E+08	7.70673178530E+06	6.11109717680E+08
POSITION REL. TO MOON	. . .	-2.22469946017E+05	-2.90482692983E+05	-2.53921437851E+04	3.66766728002E+05

```
SPACECRAFT PIERCED SPHERE OF INFLUENCE OF MARS       AT DATE. . . . 2442092.30264093

     POSITION. . . . .  -3.84633772932E+05  -3.18001498213E+05   2.65741620385E+05   5.65408437283E+05
     VELOCITY. . . . .   1.97432216590E+00   1.60030774455E+00  -1.39516068074E+00   2.89920786029E+00

     B . . . . .    8.75739013819E+03    B.T . . . . .    4.88624989869E+03    B.R . . . . .    7.26749227450E+03


INTERPOLATED INFORMATION AT SPHERE OF INFLUENCE

SPACECRAFT PIERCED SPHERE OF INFLUENCE OF MARS       AT DATE. . . . 2442092.30234592

     POSITION. . . . .  -3.84684098822E+05  -3.18042290396E+05   2.65777183326E+05   5.65482330001E+05
     VELOCITY. . . . .   1.97432216590E+00   1.60030774455E+00  -1.39516068074E+00   2.89920786029E+00

     B . . . . .    8.75737960657E+03    B.T . . . . .    4.88624398159E+03    B.R . . . . .    7.26748356213E+03
```

TRAJECTORY TIME = 2.10607087512E+02   TOTAL TIME INCREMENTS = 4714

SPACECRAFT INERTIAL TRAJECTORY

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
POSITION	-1.81667761536E+07	2.36740627965E+08	5.80938061325E+06	2.37507695850E+08
VELOCITY	-2.24004136398E+01	2.68191027270E+00	1.71678843949E+00	2.26256168201E+01

****

CALENDAR DATE = FEBRUARY 20, 0 HR, 0 MIN, 0. SEC, 1974
JULIAN DATE = 2442098.50000000

EPHEMERIS DATA

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
POSITION OF SUN	0.	0.	0.	0.
VELOCITY OF SUN	0.	0.	0.	0.
POSITION OF EARTH	-1.29334653733E+08	7.17776403603E+07	0.	1.47917146781E+08
VELOCITY OF EARTH	-1.49415442481E+01	-2.61544209612E+01	0.	3.01214787209E+01
POSITION OF MARS	-1.84602032562E+07	2.35852769482E+08	5.41380518927E+06	2.36636043880E+08
VELOCITY OF MARS	-2.32183747818E+01	1.57278703144E-01	5.72316494526E-01	2.32259598352E+01
POSITION OF JUPITER	6.16450888550E+08	-4.31435013128E+08	-1.20794187535E+07	7.52524937065E+08
VELOCITY OF JUPITER	7.34013612218E+00	1.13262554008E+01	-2.10142325751E-01	1.34983635858E+01
POSITION OF MOON	-1.29089997366E+08	7.14566069268E+07	2.70654512588E+04	1.47547534124E+08
VELOCITY OF MOON	-1.41877159619E+01	-2.55540024759E+01	7.02734528632E-02	2.92284666911E+01

****

SPACECRAFT RELATIVE TRAJECTORIES

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
POSITION REL. TO SUN . . .	-1.81667761536E+07	2.36740627965E+08	5.80938061325E+06	2.37507695850E+08
VELOCITY REL. TO SUN . . .	-2.24004136398E+01	2.68191027270E+00	1.71678843949E+00	2.26256168201E+01
POSITION REL. TO EARTH . . .	1.11167877579E+08	1.64962987604E+08	5.80938061325E+06	1.99009630892E+08
VELOCITY REL. TO EARTH . . .	-7.45886939164E+00	2.88363312339E+01	1.71678843949E+00	2.98348134598E+01
POSITION REL. TO MARS . . .	2.93427102538E+05	8.87858482879E+05	3.95575423971E+05	1.01531870178E+06
VELOCITY REL. TO MARS . . .	8.17961142002E-01	2.52463156956E+00	1.14447194496E+00	2.89009360136E+00
POSITION REL. TO JUPITER . . .	-6.34617664704E+08	6.68175641093E+08	1.78887993668E+07	9.21693157644E+08
VELOCITY REL. TO JUPITER . . .	-2.97405497620E+01	-8.64434512815E+00	1.92693076524E+00	3.10312433688E+01
POSITION REL. TO MOON . . .	1.10923221212E+08	1.65284021038E+08	5.78231516199E+06	1.99138654669E+08
VELOCITY REL. TO MOON . . .	-8.21269767790E+00	2.82359127486E+01	1.64651498662E+00	2.94520998148E+01

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
VIRTUAL MASS DATA				
VIRTUAL MASS POSITION	-1.48684771444E+07	1.89985924078E+08	4.36096113243E+06	1.90616738359E+08
VIRTUAL MASS VELOCITY	5.84093482058E+00	-3.13304241006E+02	-6.73358698127E+00	3.13431021351E+02
SPACECRAFT POS. REL. TO V.M.	-3.29829900920E+06	4.67547038864E+07	1.44841948081E+06	4.68932727676E+07
SPACECRAFT VEL. REL. TO V.M.	-2.82413484604E+01	3.15986151278E+02	8.45037542076E+00	3.17358205199E+02
KEPLER (ANG. MOM.) VECTOR	-6.25162776185E+07	-1.29807618907E+07	2.76656250032E+08	2.83928628054E+08
ECCENTRICITY VECTOR	1.68265039282E+01	3.96056137441E-01	3.82088422443E+00	1.72594105104E+01

V.M. MAGN. = 5.25266169575E+09
V.M. MAGN. RATE = 7.04373497602E+04

	X - COMP.	Y - COMP.	Z - COMP.	RESULTANT
V.M. RELATIVE POSITIONS				
POSITION REL. TO SUN . . .	-1.48684771444E+07	1.89985924078E+08	4.36096113243E+06	1.90616738359E+08
POSITION REL. TO EARTH . . .	1.14466176588E+08	1.18208283718E+08	4.36096113243E+06	1.64604744477E+08
POSITION REL. TO MARS . . .	3.59172611173E+06	-4.58668454035E+07	-1.05284405684E+06	4.60193055613E+07
POSITION REL. TO JUPITER . . .	-6.31319365695E+08	6.21420937207E+08	1.64403798860E+07	8.86001359362E+08
POSITION REL. TO MOON . . .	1.14221520222E+08	1.18529317151E+08	4.33389568117E+06	1.64664924492E+08

TOTAL CP TIME USED IN THIS INTEGRATION . . .   108.826 SEC

```
****************************************************************************************************

ORIGINAL NOMINAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT TRAJECTORY TIME  204.40943 DAYS

                                           X                   Y                   Z                 RESULTANT
POSITION RELATIVE TO TARGET PLANET   -3.8468409882E+05   -3.1804229040E+05    2.6577718333E+05    5.6548233000E+05
VELOCITY RELATIVE TO TARGET PLANET    1.9743221659E+00    1.6003077446E+00   -1.3951606807E+00    2.8992078603E+00

B =     8.7573796066E+03     B DOT T =     4.8862439816E+03     B DOT R =     7.2674835621E+03

****************************************************************************************************
```

****

ORIGINAL NOMINAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO TARGET PLANET AT 206.60907 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	2.0031341790E+03	-1.5494068331E+03	-4.3575447685E+03	5.0399606035E+03
VELOCITY RELATIVE TO TARGET PLANET	2.8091878110E+00	4.1652688168E+00	-2.3154391326E-01	5.0293750166E+00

****

SUMMARY OF TRAJECTORY MODE

PROBLEM. . 1 PAGE. . 1

VIRTUAL MASS TRAJECTORY PROBLEM 1

ACCURACY FIGURE 5.000000E-06 INDICATES TRUE ANOMALY INCREMENT IS 7.6955835381654E-03 RADIANS

INITIAL TRAJECTORY TIME 0. DAYS, JULIAN DATE 2441887.8929124773 CALENDAR DATE 7 24 9 25 47.638, 1973
FINAL TRAJECTORY TIME 210.60709 DAYS, JULIAN DATE 2442098.5000000000 CALENDAR DATE 2 20 0 0 0. , 1974

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
HELIOCENTRIC ECLIPTIC COORDINATES				
INITIAL POSITION OF VEHICLE	7.90303912E+07	-1.29799646E+08	7.56015372E+01	1.51966282E+08
INITIAL VELOCITY OF VEHICLE	3.43585796E+01	1.25084614E+01	6.20079494E+00	3.70867019E+01
FINAL POSITION OF VEHICLE	-1.81667762E+07	2.36740628E+08	5.80938061E+06	2.37507696E+08
FINAL VELOCITY OF VEHICLE	-2.24004136E+01	2.68191027E+00	1.71678844E+00	2.26256168E+01
AT FINAL TIME				
POSITION OF VEHICLE RELATIVE TO EARTH	1.11167878E+08	1.64962988E+08	5.80938061E+06	1.99009631E+08
VELOCITY OF VEHICLE RELATIVE TO EARTH	-7.45886939E+00	2.88363312E+01	1.71678844E+00	2.98348135E+01
POSITION RELATIVE TO TARGET PLANET. .	2.93427103E+05	8.87858483E+05	3.95575424E+05	1.01531870E+06
VELOCITY RELATIVE TO TARGET PLANET. .	8.17961142E-01	2.52463157E+00	1.14447194E+00	2.89009360E+00

AT CLOSEST APPROACH. . . . CALENDAR DATE 2 16 0 2 51.651, 1974. . .JULIAN DATE 2442094.5019866973

POSITION RELATIVE TO TARGET PLANET. .	2.00313418E+03	-1.54940683E+03	-4.35754477E+03	5.03998060E+03
VELOCITY RELATIVE TO TARGET PLANET. .	2.80918781E+00	4.16526882E+00	-2.31543913E-01	5.02937502E+00

AT SPHERE OF INFLUENCE. . . . CALENDAR DATE 2 13 19 15 22.687, 1974. . .JULIAN DATE 2442092.3023459166

POSITION RELATIVE TO TARGET PLANET. .	-3.84684099E+05	-3.18042290E+05	2.65777183E+05	5.65482330E+05
VELOCITY RELATIVE TO TARGET PLANET. .	1.97432217E+00	1.60030774E+00	-1.39516068E+00	2.89920786E+00

B = 8.75737961E+03 B DOT T = 4.88624398E+03 B DOT R = 7.26748356E+03

TOTAL NUMBER OF TIME INCREMENTS FOR THIS PROBLEM IS 4714

TOTAL CP TIME FOR THIS PROBLEM IS 108.828 SEC

INPUT DATA PROBLEM....2

NUMERICAL DIFFERENCING PROCEDURE

LEVL	ITER	STEP	ACCY	XDOT	YDOT	ZDOT	TRAJECTORY B.T OR INCL	TRAJECTORY B.R OR RCA	TRAJECTORY TSI OR TCA	TARGET B.T/INCL	TARGET B.R/RCA	TARGET TSI/TCA	TIME PER INTEG (SEC)	TOTAL CP TIME (SEC)	NO OF INTEG INCR
										STATE TRANSITION MATRIX					
TARGETING AND CONSTRUCTION OF SPHERE-OF-INFLUENCE STATE TRANSITION MATRIX															
1	0	0	5.00E-04	34.386674	12.518986	6.164171	-160137.77	122307.69	27072.827	5074.68	6930.76	27072.348	5.86	6.1	253
1	0	1	5.00E-04	34.386684	12.518986	6.164171	-159992.04	121656.42	27072.822	-1.25E-07	5.31E-07	-7.92E-02			
1	0	2	5.00E-04	34.386674	12.518996	6.164171	-160251.97	122573.48	27072.829	-1.87E-07	8.06E-09	-6.88E-03			
1	0	3	5.00E-04	34.386674	12.518986	6.164181	-160046.14	121883.85	27072.824	7.48E-08	-8.35E-07	1.17E-01			
1	1	0	5.00E-04	34.342614	12.490438	6.216694	-20272.15	148999.45	27073.464	5112.11	6945.06	27072.274	5.97	29.7	258
1	1	1	5.00E-04	34.342624	12.490438	6.216694	-20132.54	148326.10	27073.459	-3.23E-08	5.29E-07	-7.74E-02			
1	1	2	5.00E-04	34.342614	12.490448	6.216694	-20382.10	149259.00	27073.466	-1.76E-07	3.06E-09	-5.55E-03			
1	1	3	5.00E-04	34.342614	12.490438	6.216704	-20186.61	148574.29	27073.461	-5.60E-08	-8.59E-07	1.19E-01			
1	2	0	5.00E-04	34.358787	12.492152	6.195468	10751.43	744.36	27072.307	5099.87	6929.76	27072.256	6.28	53.9	271
1	2	1	5.00E-04	34.358797	12.492152	6.195468	10917.11	92.12	27072.303	-1.58E-08	4.20E-07	-7.34E-02			
1	2	2	5.00E-04	34.358787	12.492162	6.195468	10631.94	994.84	27072.308	-1.74E-07	-3.27E-09	-6.84E-03			
1	2	3	5.00E-04	34.358787	12.492152	6.195478	10851.45	337.37	27072.304	-8.19E-08	-7.00E-07	1.13E-01			
1	3	0	5.00E-04	34.365213	12.493463	6.185825	4867.09	6621.97	27072.254	5098.80	6930.70	27072.260	6.28	79.1	271
1	3	1	5.00E-04	34.365223	12.493463	6.185825	5031.29	5972.44	27072.250	-1.93E-08	4.25E-07	-7.34E-02			
1	3	2	5.00E-04	34.365213	12.493473	6.185825	4744.13	6880.87	27072.256	-1.70E-07	-5.92E-09	-6.10E-03			
1	3	3	5.00E-04	34.365213	12.493463	6.185835	4966.08	6218.22	27072.252	-7.78E-08	-7.12E-07	1.14E-01			
1	4	0	5.00E-04	34.364914	12.493387	6.186248	5076.49	6968.35	27072.260	5098.90	6930.74	27072.259	6.27	104.3	271
TARGETING TO SPHERE-OF-INFLUENCE CONDITIONS															
2	0	0	2.50E-05	34.364914	12.493387	6.186248	122071.78	-187536.25	27071.252	4897.54	6867.20	27072.439	23.90	128.2	1035
3	0	0	5.00E-06	34.362602	12.504879	6.192522	28737.57	-8840.80	27072.152	4947.03	6895.41	27072.282	50.51	178.7	2193
3	1	0	5.00E-06	34.360244	12.508034	6.197948	8416.10	3707.69	27072.259	4950.82	6896.29	27072.300	50.54	229.3	2193
3	2	0	5.00E-06	34.358626	12.508351	6.200676	5686.19	6244.80	27072.296	4952.16	6896.57	27072.301	50.53	279.8	2193
3	3	0	5.00E-06	34.358536	12.508440	6.200861	5067.57	6829.07	27072.300	4952.34	6896.64	27072.301	50.57	330.4	2193
3	4	0	5.00E-06	34.358546	12.508457	6.200855	4962.85	6893.56	27072.301	4952.35	6896.64	27072.301	50.57	381.0	2193
CONSTRUCTION OF CLOSEST-APPROACH STATE TRANSITION MATRIX															
1	0	0	5.00E-04	34.364914	12.493387	6.186248	37.05	4887.46	27074.501	38.00	4800.00	27074.500	9.28	390.3	402
1	0	1	5.00E-04	34.364915	12.493387	6.186248	36.74	4862.01	27074.500	6.16E-06	9.31E-08	-1.17E-02			
1	0	2	5.00E-04	34.364914	12.493388	6.186248	37.21	4898.07	27074.501	2.34E-05	-1.01E-08	-1.56E-02			
1	0	3	5.00E-04	34.364914	12.493387	6.186249	36.81	4860.25	27074.501	3.35E-06	-1.28E-07	4.85E-03			
TARGETING TO CLOSEST-APPROACH CONDITIONS															
1	0	0	5.00E-06	34.358546	12.508457	6.200855	37.02	4817.95	27074.500	38.00	4800.00	27074.500	78.33	496.5	3399
1	1	0	5.00E-06	34.358546	12.508475	6.200862	38.11	4846.91	27074.500	38.00	4800.00	27074.500	78.31	574.9	3398
1	2	0	5.00E-06	34.358547	12.508480	6.200865	37.94	4788.42	27074.500	38.00	4800.00	27074.500	78.30	653.2	3400

```
I N P U T   D A T A   F O R   P R O B L E M . . . .  3

MODE TO BE EXECUTED. . .ERROR ANALYSIS MODE

 LAUNCH DATE              7 24  9 25 47.639 1973     JULIAN DATE . . . 2441887.89291248

 FINAL DATE               2 20  0  0  0.    1974     JULIAN DATE . . . 2442098.50000000

 INITIAL TRAJECTORY TIME =    0.

 AUGMENTATION CODE. . . . .  1

 INITIAL STATE VECTOR
   GEOCENTRIC ECLIPTIC COORDINATES
        -1.09005292E+03
        -6.55005114E+03
         7.56015372E+01
         9.40152732E+00
        -2.87640366E+00
         6.20079494E+00

 INITIAL STATE VECTOR
   HELIOCENTRIC ECLIPTIC COORDINATES
         7.90303912E+07
        -1.29799646E+08
         7.56015372E+01
         3.43585796E+01
         1.25084614E+01
         6.20079494E+00

 NOMINAL TRAJECTORY CODE. . . 2

 NOMINAL TRAJECTORY INFORMATION

   BODIES TO BE CONSIDERED
     SUN
     EARTH
     MARS
     JUPITER
     MOON

   TARGET PLANET. . .MARS

 UNITS
   1.49598500E+08/A.U.                        8.64000000E+04/DAY

 ORBITAL ELEMENTS WILL BE CALCULATED AT EVERY TIME INTERVAL

 OUTPUT FROM VIRTUAL MASS PROGRAM WILL BE SUPPRESSED AT INITIAL AND FINAL STEPS

 VIRTUAL MASS PROGRAM WILL INTEGRATE UNTIL REACHING A NORMAL STOPPING CONDITION

 ACCURACY FIGURE. . . . . 5.00000E-06

 PRINT INTERVALS
   3.00000E+02 DAYS
       10000 INCREMENTS
```

```
MEASUREMENT SCHEDULE
  FROM      .20 DAYS TO     4.20 DAYS, EVERY     1.00 DAYS, MEASURE CODE    4
  FROM      .60 DAYS TO     4.60 DAYS, EVERY     1.00 DAYS, MEASURE CODE    6
  FROM      .90 DAYS TO     4.90 DAYS, EVERY     1.00 DAYS, MEASURE CODE    8
  FROM     5.30 DAYS TO    29.30 DAYS, EVERY     4.00 DAYS, MEASURE CODE    3
  FROM     6.70 DAYS TO    30.70 DAYS, EVERY     4.00 DAYS, MEASURE CODE    6
  FROM    34.10 DAYS TO   184.10 DAYS, EVERY    30.00 DAYS, MEASURE CODE    3
  FROM    44.50 DAYS TO   194.50 DAYS, EVERY    30.00 DAYS, MEASURE CODE    5
  FROM    54.80 DAYS TO   204.80 DAYS, EVERY    30.00 DAYS, MEASURE CODE    7
  FROM   187.30 DAYS TO   202.30 DAYS, EVERY     3.00 DAYS, MEASURE CODE    3
  FROM   203.50 DAYS TO   207.50 DAYS, EVERY     1.00 DAYS, MEASURE CODE    5
  FROM     8.00 DAYS TO    32.00 DAYS, EVERY     4.00 DAYS, MEASURE CODE    8

EVENT SCHEDULE
  TIME OF EVENT          EVENT
        2.000          PREDICTION          PREDICTING TO TIME   5.00
        4.000          PREDICTION          PREDICTING TO TIME   5.00
        4.400          PREDICTION          PREDICTING TO TIME   5.00
        4.700          PREDICTION          PREDICTING TO TIME   5.00
        5.000          EIGENVALUE
        5.001          GUIDANCE            GUIDANCE POLICY    3, EXECUTION ERROR CODE    1
       20.000          PREDICTION          PREDICTING TO TIME  30.00
       25.000          PREDICTION          PREDICTING TO TIME  30.00
       29.500          PREDICTION          PREDICTING TO TIME  30.00
       30.000          EIGENVALUE
       30.001          GUIDANCE            GUIDANCE POLICY    3, EXECUTION ERROR CODE    1
       90.000          EIGENVALUE
      197.000          EIGENVALUE
      199.410          PREDICTION          PREDICTING TO TIME 204.41
      201.410          PREDICTION          PREDICTING TO TIME 204.41
      203.410          PREDICTION          PREDICTING TO TIME 204.41
      206.608          EIGENVALUE

  FOR EIGENVALUE EVENTS, THE SIGMA LEVEL OF THE HYPERELLIPSOID IS K = 3

INITIAL COVARIANCE MATRIX
        1.00000000E+00
        0.                 1.00000000E+00
        0.                 0.                 1.00000000E+00
        0.                 0.                 0.                 9.00000000E-06
        0.                 0.                 0.                 0.                 9.00000000E-06
        0.                 0.                 0.                 0.                 0.                 9.00000000E-06
```

```
STATE TRANSITION MATRIX CODE . . . 1

DYNAMIC NOISE IS ZERO

MEASUREMENT NOISE IS CONSTANT
  RANGE (EARTH-CENTERED. . . . .  9.000000000000E-06
  RANGE-RATE (EARTH-CENTERED . .  9.000000000000E-12
  RANGE (STATION NUMBER 1) . . .  9.000000000000E-06
  RANGE-RATE (STATION NUMBER 1).  9.000000000000E-12
  RANGE (STATION NUMBER 2) . . .  9.000000000000E-06
  RANGE-RATE (STATION NUMBER 2).  9.000000000000E-12
  RANGE (STATION NUMBER 3) . . .  9.000000000000E-06
  RANGE-RATE (STATION NUMBER 3).  9.000000000000E-12
  STAR PLANET ANGLE NUMBER 1 . .  1.000000000000E-08
  STAR PLANET ANGLE NUMBER 2 . .  1.000000000000E-08
  STAR PLANET ANGLE NUMBER 3 . .  1.000000000000E-08
  APPARENT PLANET DIAMETER . . .  1.000000000000E-08

STATION LOCATION CONSTANTS
  STATION NO. 1     ALTITUDE =  1.03100000E+00     LATITUDE =  3.53840000E+01     LONGITUDE = -1.16833000E+02
  STATION NO. 2     ALTITUDE =  5.00000000E-02     LATITUDE =  4.04170000E+01     LONGITUDE = -3.66700000E+00
  STATION NO. 3     ALTITUDE =  5.00000000E-02     LATITUDE = -3.53110000E+01     LONGITUDE =  1.49136000E+02
```

ERROR ANALYSIS MODE AT TRAJECTORY TIME .200 DAYS

PROBLEM. . 301 PAGE. . 1

INITIAL TRAJECTORY TIME 0. DAYS, CALENDAR DATE 7 24 9 25 47.638, 1973, JULIAN DATE 2441887.8929124773

FINAL TRAJECTORY TIME .20000 DAYS, CALENDAR DATE 7 24 14 13 47.638, 1973, JULIAN DATE 2441888.0929124802

STATE VECTOR

	INITIAL	FINAL
RX	7.90303912E+07	7.95412021E+07
RY	-1.29799646E+08	-1.29487880E+08
RZ	7.56015372E+01	4.47920792E+04
VX	3.43585796E+01	2.84817434E+01
VY	1.25084614E+01	1.80411774E+01
VZ	6.20079494E+00	1.93516731E+00

AT INITIAL TRAJECTORY TIME 0. DAYS

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
POSITION OF VEHICLE RELATIVE TO EARTH	-1.09005292E+03	-6.55005114E+03	7.56015372E+01	6.64056490E+03
VELOCITY OF VEHICLE RELATIVE TO EARTH	9.40152732E+00	-2.87640356E+00	6.20079494E+00	1.16237804E+01
POSITION RELATIVE TO TARGET PLANET. .	-1.06945275E+08	-3.99078856E+07	6.44858919E+06	1.14330728E+08
VELOCITY RELATIVE TO TARGET PLANET. .	2.28703913E+01	-1.13684333E+01	5.97999407E+00	2.62308293E+01

AT FINAL TRAJECTORY TIME .200 DAYS

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
POSITION OF VEHICLE RELATIVE TO EARTH	7.88917704E+04	3.86220416E+04	4.47920792E+04	9.85997155E+04
VELOCITY OF VEHICLE RELATIVE TO EARTH	3.57648455E+00	2.57157052E+00	1.93516731E+00	4.81135003E+00
POSITION RELATIVE TO TARGET PLANET. .	-1.06632558E+08	-4.00089057E+07	6.48947595E+06	1.14075976E+08
VELOCITY RELATIVE TO TARGET PLANET. .	1.70419017E+01	-5.85901967E+00	1.71269152E+00	1.81021501E+01

STATISTICAL DATA AFTER MEASUREMENT 1

RANGE AND RANGE-RATE WERE MEASURED FROM STATION 1 AT TRAJECTORY TIME .20000 DAYS

ERROR ANALYSIS MODE AT TRAJECTORY TIME .200 DAYS PROBLEM. . 301 PAGE. . 2

STATE TRANSITION MATRIX -- PSI(.200, 0.

-3.70144241E+00	-2.57474328E+01	6.35323037E+00	2.68650424E+04	-1.27340014E+04	1.36606325E+04
-5.46519237E+00	1.25540399E+01	-5.91312567E+00	-3.73081281E+03	1.47977512E+04	-3.19020528E+03
4.01211207E+00	-1.85740485E+01	-4.46965075E+00	1.26214214E+04	-8.81032784E+03	1.62316492E+04
-2.71129832E-04	-1.82568484E-03	3.73832601E-04	1.85160827E+00	-8.62712089E-01	1.05649382E+00
-2.96276998E-04	3.95029868E-04	-2.91998373E-04	-5.49893260E-02	6.49446416E-01	-7.25063267E-02
1.97297632E-04	-1.24671118E-03	-2.65343063E-04	9.63260797E-01	-5.76716680E-01	9.86546564E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

OBSERVATION MATRIX

7.67206944E-01	3.89084589E-01	5.09908510E-01	0.	0.	0.
5.24661906E-07	4.63682127E-06	-4.32752133E-06	7.67206944E-01	3.89084589E-01	5.09908510E-01

MEASUREMENT NOISE MATRIX

9.00000000E-06	0.
0.	9.00000000E-12

K MATRIX

-5.09099774E+00	8.60176489E+04
-7.41200752E-01	6.49140838E+03
1.01866041E+01	-1.34375105E+05
-2.20540938E-04	4.11524837E+00
-5.13712027E-05	6.07222055E-01
4.69523014E-04	-5.98291826E+00

ERROR ANALYSIS MODE AT TRAJECTORY TIME .200 DAYS PROBLEM. . 301 PAGE. . 3

COVARIANCE MATRIX AT TIME .20000 DAYS, JUST BEFORE THE MEASUREMENT

1.03514781E+04	-3.33076191E+03	6.49198984E+03	7.26840483E-01	-1.07569906E-01	4.49972771E-01
-3.33076191E+03	2.41006562E+03	-2.29186814E+03	-2.31050176E-01	9.87263573E-02	-1.52636800E-01
6.49198984E+03	-2.29186814E+03	4.88456741E+03	4.64225729E-01	-7.55558587E-02	3.24402596E-01
7.26840483E-01	-2.31050176E-01	4.64225729E-01	5.11465279E-05	-7.39838707E-06	3.20340366E-05
-1.07569906E-01	9.87263573E-02	-7.55558587E-02	-7.39838707E-06	4.19964647E-06	-4.96488176E-06
4.49972771E-01	-1.52636800E-01	3.24402596E-01	3.20340366E-05	-4.96488176E-06	2.17673506E-05

COVARIANCE MATRIX AT TIME .20000 DAYS, AFTER CONSIDERING THE MEASUREMENT

2.13078830E+02	-5.41504480E+02	9.25959135E+01	1.25897960E-02	-2.54925343E-02	6.00171076E-03
-5.41504480E+02	1.62593830E+03	-4.25846267E+02	-3.34601258E-02	7.54169086E-02	-2.50439658E-02
9.25959135E+01	-4.25846267E+02	1.85621698E+02	6.58910042E-03	-1.91907133E-02	1.00795948E-02
1.25897960E-02	-3.34601258E-02	6.58910042E-03	7.57454261E-07	-1.56905529E-06	4.04907555E-07
-2.54925343E-02	7.54169086E-02	-1.91907133E-02	-1.56905529E-06	3.50368287E-06	-1.13510837E-06
6.00171076E-03	-2.50439658E-02	1.00795948E-02	4.04907555E-07	-1.13510837E-06	5.63918073E-07

ERROR ANALYSIS MODE -- PREDICTION EVENT AT TRAJECTORY TIME 2.000 DAYS, PREDICTING TO TRAJECTORY TIME 5.000 DAYS
PROBLEM. 301 PAGE. 19

****

STATE VECTOR AT TIME 2.000 DAYS

RX	8.3856355945202E+07
RY	-1.2666722269047E+08
RZ	3.0232510029620E+05
VX	2.7378266441675E+01
VY	1.8432895396780E+01
VZ	1.5855411547415E+00

STATE TRANSITION MATRIX -- PSI(2.000, 1.900)

1.00002644E+00	4.63985657E-05	3.56369927E-05	8.64007359E+03	1.29259622E-01	9.92869362E-02
4.63985528E-05	9.99993500E-01	2.51658762E-05	1.29259478E-01	8.63998181E+03	7.01094851E-02
3.56369944E-05	2.51658844E-05	9.99980063E-01	9.92869543E-02	7.01095760E-02	8.63994439E+03
5.97473724E-09	1.04934462E-08	8.05541841E-09	1.00002518E+00	4.42642330E-05	3.39613673E-05
1.04934467E-08	-1.46467841E-09	5.69107073E-09	4.42642217E-05	9.99993845E-01	2.40046454E-05
8.05541834E-09	5.69107038E-09	-4.50937806E-09	3.39613687E-05	2.40046525E-05	9.99980975E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX AT TIME OF PREDICTION EVENT -- P(2.000, 1.900)

3.02219926E-01	-1.56571863E-01	-3.63538827E-01	3.82145476E-06	-2.42035713E-06	-4.09137031E-06
-1.56571863E-01	1.25176524E-01	1.29852083E-01	-1.79974767E-06	1.36645715E-06	1.60552410E-06
-3.63538827E-01	1.29852083E-01	5.14987245E-01	-4.83889319E-06	2.76489808E-06	5.60672483E-06
3.82145476E-06	-1.79974767E-06	-4.83889319E-06	5.01786442E-11	-3.10128281E-11	-5.48851629E-11
-2.42035713E-06	1.36645715E-06	2.76489808E-06	-3.10128281E-11	2.03634371E-11	3.22494709E-11
-4.09137031E-06	1.60552410E-06	5.60672483E-06	-5.48851629E-11	3.22494709E-11	6.23862361E-11

STATE TRANSITION MATRIX -- PSI(5.000, 2.000)

1.00898032E+00	1.60480986E-02	1.21437977E-02	2.59670327E+05	8.52548910E+02	6.39966329E+02
1.60491017E-02	9.98023974E-01	8.68653719E-03	8.52572155E+02	2.59100660E+05	4.60923647E+02
1.21436718E-02	8.68590422E-03	9.93116674E-01	6.39963413E+02	4.60908979E+02	2.58832575E+05
4.92534735E-08	8.82134407E-08	6.65025412E-08	1.00372312E+00	4.78930643E-03	5.07226133E-03
8.82250140E-08	-1.03668671E-08	4.77258974E-08	6.78962465E-03	9.99244857E-01	3.66815249E-03

****

6.65010892E-08	4.77185947E-08	-3.77051180E-08	5.07222141E-03	3.66795170E-03	9.97074976E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX AT PREDICTION TIME -- P(5.000, 2.000)

5.59314300E+00	-3.30087570E+00	-6.29745602E+00	1.65374140E-05	-1.03039838E-05	-1.80022576E-05
-3.30087570E+00	2.18045177E+00	3.39717310E+00	-9.67639387E-06	6.54488664E-06	9.80783554E-06
-6.29745602E+00	3.39717310E+00	7.53078610E+00	-1.87403389E-05	1.09702731E-05	2.13986892E-05
1.65374140E-05	-9.67639387E-06	-1.87403389E-05	4.89968091E-11	-3.03571416E-11	-5.35968012E-11
-1.03039838E-05	6.54488664E-06	1.09702731E-05	-3.03571416E-11	1.99588922E-11	3.15975486E-11
-1.80022576E-05	9.80783554E-06	2.13986892E-05	-5.35968012E-11	3.15975486E-11	6.08964022E-11

POSITION EIGENVALUES

1	2.1705605108E-04
2	5.8700306265E-01
3	1.4717160755E+01

POSITION EIGENVECTORS

1	7.5154575506E-01	-2.3965791821E-01	-6.1460805420E-01
2	5.3372257325E-01	7.6846693637E-01	3.5298552735E-01
3	3.8771019184E-01	-5.9331496691E-01	7.0544890474E-01

```
ERROR ANALYSIS MODE -- PREDICTION EVENT AT TRAJECTORY TIME       2.000 DAYS, PREDICTING TO TRAJECTORY TIME      5.000 DAYS
                                                                                    PROBLEM. .       301      PAGE. .       21

******************************************************************************************************************************

                    FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                         THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

    2.602E+03 X**2 +  1.313E+03 Y**2 +  6.932E+02 Z**2 +  3.695E+03 XY +  2.685E+03 XZ +  1.905E+03 YZ =   9

          XY HYPERELLIPSOID. . . . . .  2.602E+03 X**2 +  3.695E+03 XY +  1.313E+03 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  2.602E+03 X**2 +  2.685E+03 XZ +  6.932E+02 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  1.313E+03 Y**2 +  1.905E+03 YZ +  6.932E+02 Z**2 =   9


          VELOCITY EIGENVALUES
             1     2.2846772017E-15
             2     3.0294927992E-12
             3     1.2682032594E-10


          VELOCITY EIGENVECTORS
             1     7.5154148881E-01    -2.2370238246E-01    -6.2059860996E-01
             2     5.3737589764E-01     7.5326444747E-01     3.7923583272E-01
             3     3.8263890974E-01    -6.1850619743E-01     6.8632175290E-01

                    FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                         THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

    2.472E+14 X**2 +  1.266E+14 Y**2 +  6.421E+13 Z**2 +  3.534E+14 XY +  2.518E+14 XZ +  1.797E+14 YZ =   9

          XY HYPERELLIPSOID. . . . . .  2.472E+14 X**2 +  3.534E+14 XY +  1.266E+14 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  2.472E+14 X**2 +  2.518E+14 XZ +  6.421E+13 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  1.266E+14 Y**2 +  1.797E+14 YZ +  6.421E+13 Z**2 =   9
```

ERROR ANALYSIS MODE -- PREDICTION EVENT AT TRAJECTORY TIME 2.000 DAYS, PREDICTING TO TRAJECTORY TIME 5.000 DAYS
PROBLEM. • 301 PAGE. • 22

****

CORRELATION COEFFICIENT MATRIX AT PREDICTION TIME-- 2.000 DAYS

1.00000000E+00	-9.45209256E-01	-9.70324563E-01	9.98977575E-01	-9.75234353E-01	-9.75445913E-01
-9.45209256E-01	1.00000000E+00	8.38347894E-01	-9.36173614E-01	9.92111925E-01	8.51146103E-01
-9.70324563E-01	8.38347894E-01	1.00000000E+00	-9.75603583E-01	8.94806367E-01	9.99243192E-01
9.98977575E-01	-9.36173614E-01	-9.75603583E-01	1.00000000E+00	-9.70753046E-01	-9.81203648E-01
-9.75234353E-01	9.92111925E-01	8.94806367E-01	-9.70753046E-01	1.00000000E+00	9.06336262E-01
-9.75445913E-01	8.51146103E-01	9.99243192E-01	-9.81203648E-01	9.06336262E-01	1.00000000E+00

****

STATE VECTOR AT TIME 5.000 DAYS

RX	9.0833341565653E+07
RY	-1.2173726989143E+08
RZ	7.0743040831075E+05
VX	2.6461774807615E+01
VY	1.9610429913455E+01
VZ	1.5487009617635E+00

STATE TRANSITION MATRIX -- PSI(5.000, 4.900)

1.00000010E+00	-2.03130758E-06	1.16445558E-08	8.64000249E+03	-4.60714915E-03	1.32011692E-04
-2.03085325E-06	1.00000132E+00	-1.56435069E-08	-4.60616269E-03	8.64000523E+03	2.76004439E-05
1.16556347E-08	-1.56618937E-08	9.99998587E-01	1.32035744E-04	2.75605225E-05	8.63999646E+03
2.29595305E-11	-4.70228673E-10	2.70484387E-12	1.00000010E+00	-2.03192222E-06	1.17141910E-08
-4.70228671E-10	3.04233493E-10	-3.63217528E-12	-2.03146791E-06	1.00000131E+00	-1.57201137E-08
2.70484390E-12	-3.63217533E-12	-3.27192177E-10	1.17252694E-08	-1.57384995E-08	9.99998587E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX AT TIME OF EIGENVECTOR EVENT -- P(5.000, 4.900)

8.72846122E-02	-9.34308126E-02	-3.54285043E-02	2.30210576E-07	-2.28902604E-07	-1.13146749E-07
-9.34308126E-02	1.51587143E-01	-3.23682291E-02	-2.23781225E-07	3.46801698E-07	-6.46218680E-08
-3.54285043E-02	-3.23682291E-02	1.10194969E-01	-1.24591329E-07	-4.55395573E-08	2.99341386E-07
2.30210576E-07	-2.23781225E-07	-1.24591329E-07	6.53344354E-13	-5.83711179E-13	-4.16743433E-13
-2.28902604E-07	3.46801698E-07	-4.55395573E-08	-5.83711179E-13	8.19274561E-13	-4.60826618E-14
-1.13146749E-07	-6.46218680E-08	2.99341386E-07	-4.16743433E-13	-4.60826618E-14	8.54414807E-13

```
************************************************************************************************************

POSITION EIGENVALUES
   1       2.9914501193971E-06
   2       2.1858778377471E-01
   3       1.3047594906503E-01


POSITION EIGENVECTORS
   1       7.4090990065450E-01     -5.6804467565896E-01     -3.5829843087515E-01
   2       5.4150102162068E-01      8.2084739733906E-01     -1.8162101713573E-01
   3       3.9727718622626E-01     -5.9454156601175E-02      9.1577073581053E-01

        FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
              THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

1.835E+05 X**2 +  9.802E+04 Y**2 +  5.277E+04 Z**2 +  2.682E+05 XY +  1.968E+05 XZ •  1.438E+05 YZ =  9

XY HYPERELLIPSOID. . . . . .  1.835E+05 X**2 +  2.682E+05 XY +  9.802E+04 Y**2 =  9

XZ HYPERELLIPSOID. . . . . .  1.835E+05 X**2 +  1.968E+05 XZ +  5.277E+04 Z**2 =  9

YZ HYPERELLIPSOID. . . . . .  9.802E+04 Y**2 +  1.438E+05 YZ +  5.277E+04 Z**2 =  9


VELOCITY EIGENVALUES
   1       1.0272347094176E-16
   2       1.4407967541798E-12
   3       8.8613424410544E-13


VELOCITY EIGENVECTORS
   1       7.3930796827901E-01     -6.7336313241503E-01      2.4124558729925E-03
   2       5.4875640613442E-01      6.0041402629880E-01     -5.8169528427695E-01
   3       3.9024368638782E-01      4.3137580939111E-01      8.1340320647669E-01
```

```
ERROR ANALYSIS MODE --   EIGENVECTOR EVENT AT TRAJECTORY TIME        5.000 DAYS
                                                                           PROBLEM. .      301     PAGE. .     64

*************************************************************************************************************

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                      THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

5.321E+15 X**2 +  2.932E+15 Y**2 +  1.483E+15 Z**2 •  7.898E+15 XY •  5.617E+15 XZ •  4.169E+15 YZ =   9

      XY HYPERELLIPSOID. . . . . .  5.321E+15 X**2 +  7.898E+15 XY +  2.932E+15 Y**2 =   9

      XZ HYPERELLIPSOID. . . . . .  5.321E+15 X**2 +  5.617E+15 XZ +  1.483E+15 Z**2 =   9

      YZ HYPERELLIPSOID. . . . . .  2.932E+15 Y**2 +  4.169E+15 YZ +  1.483E+15 Z**2 =   9


CORRELATION COEFFICIENT MATRIX AT TIME OF EIGENVECTOR EVENT  --   5.000 DAYS

      1.00000000E+00  -8.12250708E-01  -3.61246037E-01   9.64018476E-01  -8.55986974E-01  -4.14322820E-01
     -8.12250708E-01   1.00000000E+00  -2.50441832E-01  -7.11084793E-01   9.84091081E-01  -1.79561796E-01
     -3.61246037E-01  -2.50441832E-01   1.00000000E+00  -4.64339750E-01  -1.51562998E-01   9.75554415E-01
      9.64018476E-01  -7.11084793E-01  -4.64339750E-01   1.00000000E+00  -7.97833384E-01  -5.57780479E-01
     -8.55986974E-01   9.84091081E-01  -1.51562998E-01  -7.97833384E-01   1.00000000E+00  -5.50793033E-02
     -4.14322820E-01  -1.79561796E-01   9.75554415E-01  -5.57780479E-01  -5.50793033E-02   1.00000000E+00
```

****

STATE VECTOR AT TIME 5.001 DAYS

RX	9.0835627849796E+07
RY	-1.2173557553333E+08
RZ	7.0756421578223E+05
VX	2.6461469377896E+01
VY	1.9610822241448E+01
VZ	1.5486942142779E+00

STATE TRANSITION MATRIX -- PSI(5.001, 5.000)

1.00000000E+00	-4.89057329E-10	2.82396329E-12	8.64021546E+01	1.21348916E-03	9.59068020E-05
-4.54850393E-11	1.00000000E+00	-3.64819286E-13	1.21654250E-03	8.64014158E+01	7.11745896E-05
1.36373135E-11	-1.83117965E-11	1.00000000E+00	9.59812358E-05	7.10510513E-05	8.64005209E+01
2.38462003E-13	-4.70488432E-12	2.73434447E-14	1.00000000E+00	-3.61047706E-10	-1.12909682E-11
-4.70488438E-12	3.03336798E-12	-3.66456987E-14	8.25334662E-11	1.00000000E+00	1.51365587E-11
2.73434433E-14	-3.66456963E-14	-3.27190745E-12	-4.77395901E-13	-2.81064061E-12	1.00000000E+00

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT -- P(5.001, 5.000)

8.73243980E-02	-9.34699290E-02	-3.54490483E-02	2.30267486E-07	-2.28953731E-07	-1.13182635E-07
-9.34699290E-02	1.51647077E-01	-3.23777476E-02	-2.23832394E-07	3.46873384E-07	-6.46257522E-08
-3.54490483E-02	-3.23777476E-02	1.10246702E-01	-1.24627189E-07	-4.55434744E-08	2.99414847E-07
2.30267486E-07	-2.23832394E-07	-1.24627189E-07	6.53346563E-13	-5.83714624E-13	-4.16742726E-13
-2.28953731E-07	3.46873384E-07	-4.55434744E-08	-5.83714624E-13	8.19278823E-13	-4.60822064E-14
-1.13182635E-07	-6.46257522E-08	2.99414847E-07	-4.16742726E-13	-4.60822064E-14	8.54412846E-13

```
ERROR ANALYSIS MODE --  GUIDANCE EVENT  AT TRAJECTORY TIME        5.001 DAYS
                                                                              PROBLEM. .       301     PAGE. .       66

*****************************************************************************************************************************

          POSITION EIGENVALUES OF ABOVE MATRIX
             1     2.9939338516E-06
             2     2.1867720760E-01
             3     1.3053797525E-01


          POSITION EIGENVECTORS OF ABOVE MATRIX
             1     7.4090948216E-01    -5.6807349533E-01    -3.5825360173E-01
             2     5.4150253613E-01     8.2083252699E-01    -1.8168369766E-01
             3     3.9727590239E-01    -5.9384059570E-02     9.1577584094E-01

                   FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                           THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

   1.834E+05 X**2 +  9.794E+04 Y**2 +  5.272E+04 Z**2 +  2.680E+05 XY +  1.966E+05 XZ +  1.437E+05 YZ =   9

          XY HYPERELLIPSOID. . . . . .  1.834E+05 X**2 +  2.680E+05 XY +  9.794E+04 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  1.834E+05 X**2 .  1.966E+05 XZ +  5.272E+04 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  9.794E+04 Y**2 +  1.437E+05 YZ +  5.272E+04 Z**2 =   9


          VELOCITY EIGENVALUES OF ABOVE MATRIX
             1     1.0272342687E-16
             2     1.4408015378E-12
             3     8.8613397053E-13


          VELOCITY EIGENVECTORS OF ABOVE MATRIX
             1     7.3930795533E-01    -6.7336316313E-01     2.4078471650E-03
             2     5.4875643360E-01     6.0041801463E-01    -5.8169114167E-01
             3     3.9024367230E-01     4.3137021020E-01     8.1340618265E-01
```

****

FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

5.321E+15 X**2 + 2.932E+15 Y**2 + 1.483E+15 Z**2 + 7.898E+15 XY + 5.617E+15 XZ + 4.169E+15 YZ = 9

XY HYPERELLIPSOID. . . . . . 5.321E+15 X**2 + 7.898E+15 XY + 2.932E+15 Y**2 = 9

XZ HYPERELLIPSOID. . . . . . 5.321E+15 X**2 + 5.617E+15 XZ + 1.483E+15 Z**2 = 9

YZ HYPERELLIPSOID. . . . . . 2.932E+15 Y**2 + 4.169E+15 YZ + 1.483E+15 Z**2 = 9

STATE TRANSITION MATRIX -- PSI(5.001, 0.)

-1.37165510E+02	-9.14134307E+02	1.43266504E+02	9.37495559E+05	-3.99162542E+05	5.58277730E+05
-1.31609321E+02	1.30507232E+00	-1.05119086E+02	1.31378463E+05	1.91531555E+05	7.34046344E+04
5.29419869E+01	-5.93600058E+02	-9.60746749E+01	4.97038970E+05	-2.57778638E+05	4.56992663E+05
-3.31684182E-04	-2.19853342E-03	3.23162945E-04	2.25687462E+00	-9.44210002E-01	1.35736557E+00
-3.08535754E-04	-1.03620946E-04	-2.31755413E-04	3.98064230E-01	3.89528253E-01	2.33192241E-01
1.05006171E-04	-1.41156170E-03	-2.14367341E-04	1.20243087E+00	-6.04704866E-01	1.67742871E+00

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX RELATING THE TIME OF THIS GUIDANCE EVENT TO THAT OF THE LAST GUIDANCE EVENT -- P(5.001, 0.)

1.30241053E+07	7.91051288E+05	7.93757015E+06	3.13559604E+01	3.23480070E+00	1.89766244E+01
7.91051288E+05	5.62368638E+05	4.47612599E+05	1.94447359E+00	1.36102556E+00	1.09804757E+00
7.93757015E+06	4.47612599E+05	5.06545295E+06	1.91255630E+01	1.90351737E+00	1.20772657E+01
3.13559604E+01	1.94447359E+00	1.91255630E+01	7.54951099E-05	7.87926114E-06	4.57237798E-05
3.23480070E+00	1.36102556E+00	1.90351737E+00	7.87926114E-06	3.44073629E-06	4.61263794E-06
1.89766244E+01	1.09804757E+00	1.20772657E+01	4.57237798E-05	4.61263794E-06	2.88007315E-05

```
****************************************************************************************************************

POSITION EIGENVALUES OF ABOVE MATRIX
   1    1.7971356380E+07
   2    5.1837521048E+05
   3    1.6219534175E+05


POSITION EIGENVECTORS OF ABOVE MATRIX
   1    8.4982733754E-01    7.1241418467E-03   -5.2701303872E-01
   2    5.2100717676E-02    9.9387580790E-01    9.7449441754E-02
   3    5.2447975327E-01   -1.1027295717E-01    8.4425165876E-01

          FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

1.753E-06 X**2 +  1.964E-06 Y**2 +  4.433E-06 Z**2 + -6.010E-07 XY + -5.440E-06 XZ +  5.947E-07 YZ =  9

XY HYPERELLIPSOID. . . . . .  1.753E-06 X**2 + -6.010E-07 XY +  1.964E-06 Y**2 =  9

XZ HYPERELLIPSOID. . . . . .  1.753E-06 X**2 + -5.440E-06 XZ +  4.433E-06 Z**2 =  9

YZ HYPERELLIPSOID. . . . . .  1.964E-06 Y**2 +  5.947E-07 YZ +  4.433E-06 Z**2 =  9


VELOCITY EIGENVALUES OF ABOVE MATRIX
   1    1.0431358807E-04
   2    2.6323057230E-06
   3    7.9068391996E-07


VELOCITY EIGENVECTORS OF ABOVE MATRIX
   1    8.4947672095E-01   -2.5427496752E-02   -5.2701303872E-01
   2    9.0125629098E-02    9.9115133924E-01    9.7449441754E-02
   3    5.1987178376E-01   -1.3027841390E-01    8.4425165876E-01
```

```
****************************************************************************************************************************

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                      THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

3.584E+05 X**2 + 3.853E+05 Y**2 + 9.105E+05 Z**2 + -1.476E+05 XY + -1.114E+06 XZ + 1.109E+05 YZ = 9

    XY HYPERELLIPSOID. . . . . .  3.584E+05 X**2 + -1.476E+05 XY + 3.853E+05 Y**2 = 9
    XZ HYPERELLIPSOID. . . . . .  3.584E+05 X**2 + -1.114E+06 XZ + 9.105E+05 Z**2 = 9
    YZ HYPERELLIPSOID. . . . . .  3.853E+05 Y**2 + 1.109E+05 YZ + 9.105E+05 Z**2 = 9
```

```
ERROR ANALYSIS MODE --   GUIDANCE EVENT   AT TRAJECTORY TIME        5.001 DAYS
                                                                        PROBLEM. .        301     PAGE. .       70

*****************************************************************************************************************

TIME AT WHICH VEHICLE REACHES SPHERE OF INFLUENCE OF TARGET PLANET      204.409 DAYS

AT SPHERE OF INFLUENCE

POSITION   -3.8466944064E+05   -3.1804229771E+05    2.6579838948E+05
VELOCITY    1.9743251897E+00    1.6003138626E+00   -1.3951618687E+00

B =      8.7418190043E+03     B DOT T =      4.8949992881E+03     B DOT R =     7.2428158525E+03


VARIATION MATRIX
    1.8257155570E+00   -1.3954106590E-01    6.6637856740E-02    6.1840316995E+06   -8.6576871339E+06    6.0513045482E+03
   -4.3070893938E+00    2.0968670621E+00    6.4638942288E-01   -2.4781323089E+07   -6.0893938823E+06   -4.6951732020E+06
   -2.7022440918E-05    1.4159246348E-05   -5.8566220105E-06   -1.6132187215E+02   -4.0446226485E+01    1.1519063264E-01


UNCERTAINTY IN TARGET CONDITIONS BEFORE CORRECTION
    2.9689088227E+09   -1.3527524413E+10   -8.0523415887E+04
   -1.3527524413E+10    6.7267247574E+10    4.0090262137E+05
   -8.0523415887E+04    4.0090262137E+05    2.3900000804E+00


            EIGENVALUES OF ABOVE MATRIX
               1     2.3881489911E+08
               2     6.9997341500E+10
               3     6.4198514091E-04


            EIGENVECTORS OF ABOVE MATRIX
               1     9.8023652896E-01   -1.9782908606E-01   -3.9670281319E-07
               2     1.9782908606E-01    9.8023652894E-01   -6.0396261629E-06
               3     1.5836763126E-06    5.8417828312E-06    9.9999999998E-01
```

```
ERROR ANALYSIS MODE --  GUIDANCE EVENT  AT TRAJECTORY TIME        5.001 DAYS
                                                                  PROBLEM. .       301     PAGE. .      71

****************************************************************************************************************

                    FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                         THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  4.269E-09 X**2 +  5.700E-08 Y**2 +  1.558E+03 Z**2 +  9.083E-09 XY + -1.236E-03 XZ + -1.882E-02 YZ =   9

        XY HYPERELLIPSOID. . . . . .  4.269E-09 X**2 +  9.083E-09 XY +  5.700E-08 Y**2 =   9

        XZ HYPERELLIPSOID. . . . . .  4.269E-09 X**2 + -1.236E-03 XZ +  1.558E+03 Z**2 =   9

        YZ HYPERELLIPSOID. . . . . .  5.700E-08 Y**2 + -1.882E-02 YZ +  1.558E+03 Z**2 =   9


GUIDANCE MATRIX---THREE VARIABLE B-PLANE GUIDANCE POLICY
  -1.8691976048E-07   7.7859433309E-08   -3.2276926582E-08   -1.0000000000E+00   0.                   0.
   7.7342797039E-08   3.9485142465E-08   -1.5128862952E-08    0.                -1.0000000000E+00   0.
  -3.1083235156E-08  -1.5554973613E-08    3.2765137902E-07    0.                 0.                  -1.0000000000E+00
```

ERROR ANALYSIS MODE -- GUIDANCE EVENT AT TRAJECTORY TIME 5.001 DAYS PROBLEM. . 301 PAGE. . 72

****

COVARIANCE MATRIX ASSOCIATED WITH VELOCITY COMPONENTS

8.8683283179E-05	6.0431658621E-06	4.3868863290E-05
6.0431658621E-06	2.9561410832E-06	2.9344106559E-06
4.3868863290E-05	2.9344106559E-06	2.2491381842E-05

EXPECTED VALUE OF DELTA V. . . 8.6376518618E-03

STANDARD DEVIATION OF EXPECTED VALUE OF DELTA V. . . 6.2866347451E-03

EIGENVALUES OF ABOVE MATRIX

1	1.1095803891E-04
2	2.5407654072E-06
3	6.3200178431E-07

EIGENVECTORS OF ABOVE MATRIX

1	8.9334963882E-01	-3.9553936016E-02	-4.4761803914E-01
2	6.2078815210E-02	9.9743050318E-01	3.5757685000E-02
3	4.4505352882E-01	-5.9731712516E-02	8.9350964125E-01

EXPECTED VALUE OF VELOCITY CORRECTION

7.7164431710E-03	-3.4165312907E-04	-3.8663687891E-03

EXECUTION ERROR MATRIX

1.3106801354E-09	1.0713776213E-10	1.2124405264E-09
1.0713776213E-10	3.7257078542E-09	-5.3681999655E-11
1.2124405264E-09	-5.3681999655E-11	3.1229510976E-09

```
****************************************************************************************************

EIGENVALUES OF ABOVE MATRIX
   1    6.9843611874E-10
   2    3.7304514843E-09
   3    3.7304514843E-09


EIGENVECTORS OF ABOVE MATRIX
   1    8.9334963882E-01   -2.9175610068E-01    3.4176717299E-01
   2   -3.9553936016E-02    7.0655342549E-01    7.0655342549E-01
   3   -4.4761803914E-01   -6.4471748436E-01    6.1965914533E-01

          FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                  THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

1.197E+09 X**2 +  2.699E+08 Y**2 +  5.012E+08 Z**2 + -8.224E+07 XY + -9.307E+08 XZ +  4.121E+07 YZ =   9

     XY HYPERELLIPSOID. . . . . .  1.197E+09 X**2 + -8.224E+07 XY +  2.699E+08 Y**2 =   9

     XZ HYPERELLIPSOID. . . . . .  1.197E+09 X**2 + -9.307E+08 XZ +  5.012E+08 Z**2 =   9

     YZ HYPERELLIPSOID. . . . . .  2.699E+08 Y**2 +  4.121E+07 YZ +  5.012E+08 Z**2 =   9


MODIFIED COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT  -- P(      5.001,      5.001)

      8.73243980E-02  -9.34699290E-02  -3.54490483E-02   2.30267486E-07  -2.28953731E-07  -1.13182635E-07
     -9.34699290E-02   1.51647077E-01  -3.23777476E-02  -2.23832394E-07   3.46873384E-07  -6.46257522E-08
     -3.54490483E-02  -3.23777476E-02   1.10246702E-01  -1.24627189E-07  -4.55434744E-08   2.99414847E-07
      2.30267486E-07  -2.23832394E-07  -1.24627189E-07   1.31133348E-09   1.06554048E-10   1.21202378E-09
     -2.28953731E-07   3.46873384E-07  -4.55434744E-08   1.06554048E-10   3.72652713E-09  -5.37280819E-11
     -1.13182635E-07  -6.46257522E-08   2.99414847E-07   1.21202378E-09  -5.37280819E-11   3.12380551E-09
```

```
ERROR ANALYSIS MODE --  GUIDANCE EVENT  AT TRAJECTORY TIME        5.001 DAYS
                                                                      PROBLEM. .       301     PAGE. .      74
***********************************************************************************************************

      POSITION EIGENVALUES OF ABOVE MATRIX
         1    2.9939338516E-06
         2    2.1867720760E-01
         3    1.3053797525E-01


      POSITION EIGENVECTORS OF ABOVE MATRIX
         1    7.4090948216E-01   -5.6807349533E-01   -3.5825360173E-01
         2    5.4150253613E-01    8.2083252699E-01   -1.8168369766E-01
         3    3.9727590239E-01   -5.9384059570E-02    9.1577584094E-01

             FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                  THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

 1.834E+05 X**2 +  9.794E+04 Y**2 +  5.272E+04 Z**2 +  2.680E+05 XY +  1.966E+05 XZ +  1.437E+05 YZ =  9

      XY HYPERELLIPSOID. . . . . .  1.834E+05 X**2 +  2.680E+05 XY +  9.794E+04 Y**2 =   9

      XZ HYPERELLIPSOID. . . . . .  1.834E+05 X**2 +  1.966E+05 XZ +  5.272E+04 Z**2 =   9

      YZ HYPERELLIPSOID. . . . . .  9.794E+04 Y**2 +  1.437E+05 YZ +  5.272E+04 Z**2 =   9


      VELOCITY EIGENVALUES OF ABOVE MATRIX
         1    6.9950281154E-10
         2    3.7314309083E-09
         3    3.7307324057E-09


      VELOCITY EIGENVECTORS OF ABOVE MATRIX
         1    8.9340232215E-01   -2.0950045556E-01    3.9741898533E-01
         2   -3.9391956271E-02    8.4467589591E-01    5.3382666162E-01
         3   -4.4752716628E-01   -4.9257709040E-01    7.4638277408E-01
```

```
**********************************************************************************************************************

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                      THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

   1.195E+09 X**2 +  2.698E+08 Y**2 +  5.007E+08 Z**2 + -8.174E+07 XY + -9.288E+08 XZ +  4.100E+07 YZ =   9

         XY HYPERELLIPSOID. . . . . .  1.195E+09 X**2 + -8.174E+07 XY +  2.698E+08 Y**2 =   9

         XZ HYPERELLIPSOID. . . . . .  1.195E+09 X**2 + -9.288E+08 XZ +  5.007E+08 Z**2 =   9

         YZ HYPERELLIPSOID. . . . . .  2.698E+08 Y**2 +  4.100E+07 YZ +  5.007E+08 Z**2 =   9


UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION
      3.1817293525E+05   -2.3324908531E+04    1.1841642329E-01
     -2.3324908531E+04    1.3235393244E+06    7.2747632431E+00
      1.1841642329E-01    7.2747632431E+00    4.1571373106E-05


         EIGENVALUES OF ABOVE MATRIX
            1     3.1763207887E+05
            2     1.3240801809E+06
            3     1.3946187815E-06


         EIGENVECTORS OF ABOVE MATRIX
            1     9.9973126827E-01   -2.3181700389E-02   -7.7611744990E-07
            2     2.3181700385E-02    9.9973126826E-01   -5.5101242380E-06
            3     9.0364293170E-07    5.4906517706E-06    9.9999999998E-01
```

```
ERROR ANALYSIS MODE --  GUIDANCE EVENT  AT TRAJECTORY TIME        5.001 DAYS
                                                                      PROBLEM. .       301     PAGE. .       76

**************************************************************************************************************

                 FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                     THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

3.579E-06 X**2 +  2.253E-05 Y**2 +  7.170E+05 Z**2 +  6.244E-06 XY + -1.113E+00 XZ + -7.902E+00 YZ =   9

      XY HYPERELLIPSOID. . . . . .  3.579E-06 X**2 +  6.244E-06 XY +  2.253E-05 Y**2 =   9

      XZ HYPERELLIPSOID. . . . . .  3.579E-06 X**2 + -1.113E+00 XZ +  7.170E+05 Z**2 =   9

      YZ HYPERELLIPSOID. . . . . .  2.253E-05 Y**2 + -7.902E+00 YZ +  7.170E+05 Z**2 =   9
```

VIRTUAL MASS TRAJECTORY

ACCURACY FIGURE 5.000000E-06 INDICATES TRUE ANOMALY INCREMENT IS 7.6955835381654E-03 RADIANS

LENGTH UNITS 1.4959850000E+08/A.U.
TIME UNITS 8.6400000000E+04/DAY

ORBITAL ELEMENTS FOR EPHEMERIS CALCULATED AT EVERY TIME INTERVAL

INITIAL TRAJECTORY TIME 0. DAYS, JULIAN DATE 2441887.8929124773 CALENDAR DATE 7 24 9 25 47.638, 1973
FINAL TRAJECTORY TIME 210.60709 DAYS, JULIAN DATE 2442098.5000000000 CALENDAR DATE 2 20 0 0 0. , 1974

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
HELIOCENTRIC ECLIPTIC COORDINATES				
INITIAL POSITION OF VEHICLE	7.90303912E+07	-1.29799646E+08	7.56015372E+01	1.51966282E+08
INITIAL VELOCITY OF VEHICLE	3.43585796E+01	1.25084614E+01	6.20079494E+00	3.70867019E+01
FINAL POSITION OF VEHICLE	-1.81762879E+07	2.36744495E+08	5.80812239E+06	2.37512248E+08
FINAL VELOCITY OF VEHICLE	-2.24276586E+01	2.69245204E+00	1.71280342E+00	2.26535397E+01
AT FINAL TIME				
POSITION OF VEHICLE RELATIVE TO EARTH	1.11158366E+08	1.64966855E+08	5.80812239E+06	1.99007487E+08
VELOCITY OF VEHICLE RELATIVE TO EARTH	-7.48611432E+00	2.88468730E+01	1.71280342E+00	2.98515943E+01
POSITION RELATIVE TO TARGET PLANET. .	2.83915350E+05	8.91725725E+05	3.94317198E+05	1.01551403E+06
VELOCITY RELATIVE TO TARGET PLANET. .	7.90716215E-01	2.53517333E+00	1.14048693E+00	2.89016373E+00

AT CLOSEST APPROACH. . . . CALENDAR DATE 2 16 0 1 49.367, 1974. . .JULIAN DATE 2442094.5012658238

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
POSITION RELATIVE TO TARGET PLANET. .	2.01202458E+03	-1.57289799E+03	-4.30169624E+03	5.00268343E+03
VELOCITY RELATIVE TO TARGET PLANET. .	2.79576660E+00	4.18875820E+00	-2.44617752E-01	5.04200793E+00

AT SPHERE OF INFLUENCE. . . . CALENDAR DATE 2 13 19 14 33.389, 1974. . .JULIAN DATE 2442092.3017753363

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
POSITION RELATIVE TO TARGET PLANET. .	-3.84618444E+05	-3.18058119E+05	2.65853252E+05	5.65482330E+05
VELOCITY RELATIVE TO TARGET PLANET. .	1.97433257E+00	1.60033514E+00	-1.39516507E+00	2.89923218E+00

B = 8.71270246E+03 B DOT T = 4.93745390E+03 B DOT R = 7.17863032E+03

****

MISCELLANEOUS DATA FOR ERROR ANALYSIS MODE

THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY FROM THE PATCHED-CONIC TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION

IF THE TIME INTERVAL OVER WHICH THE STATE TRANSITION MATRIX WAS COMPUTED WAS GREATER THAN 8.000 DAYS
THE GOVERNING BODY WAS ASSUMED TO BE THE SUN IN THE ANALYTICAL CALCULATION

NUMBER OF MEASUREMENTS TAKEN 65

TOTAL NUMBER OF EVENTS	17
EIGENVECTOR EVENTS.	5
PREDICTION EVENTS	10
GUIDANCE EVENTS	2

```
FOR GUIDANCE EVENTS
     VARIANCE OF RESOLUTION ERROR          4.0000000000E-10
     VARIANCE OF PROPORTIONALITY ERROR     4.0000000000E-06
     VARIANCE OF POINTING ANGLE 1          5.0000000000E-05
     VARIANCE OF POINTING ANGLE 2          5.0000000000E-05

DYNAMIC NOISE IS ZERO

MEASUREMENT NOISE WAS CONSTANT AS SHOWN BY THE FOLLOWING NUMBERS
     RANGE (EARTH-CENTERED. . . . .    9.0000000000E-06
     RANGE-RATE (EARTH-CENTERED . .    9.0000000000E-12
     RANGE (STATION NUMBER 1) . . .    9.0000000000E-06
     RANGE-RATE (STATION NUMBER 1).    9.0000000000E-12
     RANGE (STATION NUMBER 2) . . .    9.0000000000E-06
     RANGE-RATE (STATION NUMBER 2).    9.0000000000E-12
     RANGE (STATION NUMBER 3) . . .    9.0000000000E-06
     RANGE-RATE (STATION NUMBER 3).    9.0000000000E-12
     STAR PLANET ANGLE NUMBER 1 . .    1.0000000000E-08
     STAR PLANET ANGLE NUMBER 2 . .    1.0000000000E-08
     STAR PLANET ANGLE NUMBER 3 . .    1.0000000000E-08
     APPARENT PLANET DIAMETER . . .    1.0000000000E-08

DIRECTION COSINES FOR THREE STAR PLANET ANGLES
     1   -6.1351000000E-02   2.3788600000E-01   -9.6935500000E-01
     2    2.8986000000E-02   9.6038800000E-01   -2.7714100000E-01
     3    2.0196300000E-01   8.3134300000E-01   -5.1778400000E-01
```

****

S U M M A R Y   O F   E R R O R   A N A L Y S I S   M O D E

STATISTICAL DATA FOR ERROR ANALYSIS MODE

STATE VECTOR

	INITIAL	FINAL
RX	7.9030391204E+07	-1.8176287906E+07
RY	-1.2979964568E+08	2.3674449521E+08
RZ	7.5601537160E+01	5.8081223874E+06
VX	3.4358579604E+01	-2.2427658567E+01
VY	1.2508461423E+01	2.6924520351E+00
VZ	6.2007949383E+00	1.7128034244E+00

INITIAL COVARIANCE MATRIX

1.0000000000E+00	0.	0.	0.	0.	0.
0.	1.0000000000E+00	0.	0.	0.	0.
0.	0.	1.0000000000E+00	0.	0.	0.
0.	0.	0.	9.0000000000E-06	0.	0.
0.	0.	0.	0.	9.0000000000E-06	0.
0.	0.	0.	0.	0.	9.0000000000E-06

FINAL COVARIANCE MATRIX

4.9478392309E+04	-3.6436018012E+04	4.6735107498E+04	1.4158442123E-01	-1.0445815344E-01	1.3310856663E-01
-3.6436018012E+04	2.6833458088E+04	-3.4419925206E+04	-1.0426323146E-01	7.6928592649E-02	-9.8033031285E-02
4.6735107498E+04	-3.4419925206E+04	4.4153775103E+04	1.3373442758E-01	-9.8678737864E-02	1.2575612338E-01
1.4158442123E-01	-1.0426323146E-01	1.3373442758E-01	4.0514958216E-07	-2.9891139024E-07	3.8089567164E-07
-1.0445815344E-01	7.6928592649E-02	-9.8678737864E-02	-2.9891139024E-07	2.2054624029E-07	-2.8105144447E-07
1.3310856663E-01	-9.8033031285E-02	1.2575612338E-01	3.8089567164E-07	-2.8105144447E-07	3.5817110618E-07

```
I N P U T   D A T A   F O R   P R O B L E M . . . .  4

  MODE TO BE EXECUTED. . .SIMULATION MODE

   LAUNCH DATE              7 24  9 25 47.637 1973     JULIAN DATE . . . 2441887.89291246

   FINAL DATE               2 20  0  0  0.    1974     JULIAN DATE . . . 2442098.50000000

   INITIAL TRAJECTORY TIME =    0.

   AUGMENTATION CODE. . . . .  1

   INITIAL STATE VECTOR
     HELIOCENTRIC ECLIPTIC COORDINATES
           7.90303493E+07
          -1.29799671E+08
           7.56015372E+01
           3.43165350E+01
           1.25031240E+01
           6.21517900E+00

   NOMINAL TRAJECTORY CODE. . . 2

   NOMINAL TRAJECTORY INFORMATION

     BODIES TO BE CONSIDERED
       SUN
       EARTH
       MARS
       JUPITER
       MOON

     TARGET PLANET. . .MARS

   UNITS
      1.49598500E+08/A.U.                    8.64000000E+04/DAY

   ORBITAL ELEMENTS WILL BE CALCULATED AT EVERY TIME INTERVAL

   OUTPUT FROM VIRTUAL MASS PROGRAM WILL BE SUPPRESSED AT INITIAL AND FINAL STEPS

   VIRTUAL MASS PROGRAM WILL INTEGRATE UNTIL REACHING A NORMAL STOPPING CONDITION

   ACCURACY FIGURE. . . . . 5.00000E-06

   PRINT INTERVALS
      3.00000E+02 DAYS
          10000 INCREMENTS
```

```
MEASUREMENT SCHEDULE
  FROM     .20 DAYS TO    4.20 DAYS, EVERY    2.00 DAYS, MEASURE CODE    4
  FROM    1.30 DAYS TO    3.30 DAYS, EVERY    2.00 DAYS, MEASURE CODE    4
  FROM     .60 DAYS TO    4.60 DAYS, EVERY    2.00 DAYS, MEASURE CODE    6
  FROM    1.70 DAYS TO    3.70 DAYS, EVERY    2.00 DAYS, MEASURE CODE    6
  FROM     .90 DAYS TO    4.90 DAYS, EVERY    2.00 DAYS, MEASURE CODE    8
  FROM    1.00 DAYS TO    3.00 DAYS, EVERY    2.00 DAYS, MEASURE CODE    8
  FROM    5.30 DAYS TO   29.30 DAYS, EVERY    3.00 DAYS, MEASURE CODE    3
  FROM    6.60 DAYS TO   27.60 DAYS, EVERY    3.00 DAYS, MEASURE CODE    5
  FROM    7.90 DAYS TO   28.90 DAYS, EVERY    3.00 DAYS, MEASURE CODE    7
  FROM   30.30 DAYS TO  191.30 DAYS, EVERY   23.00 DAYS, MEASURE CODE    3
  FROM   37.60 DAYS TO  198.60 DAYS, EVERY   23.00 DAYS, MEASURE CODE    5
  FROM   44.90 DAYS TO  182.90 DAYS, EVERY   23.00 DAYS, MEASURE CODE    7
  FROM  200.30 DAYS TO  207.30 DAYS, EVERY    1.00 DAYS, MEASURE CODE    3

EVENT SCHEDULE
  TIME OF EVENT          EVENT
        2.500          Q-LINEAR
        3.000          PREDICTION          PREDICTING TO TIME   5.00
        4.000          PREDICTION          PREDICTING TO TIME   5.00
        5.000          EIGENVALUE
        5.001          GUIDANCE            GUIDANCE POLICY    3, EXECUTION ERROR CODE    1
        5.002          Q-LINEAR
       30.000          EIGENVALUE
       30.001          GUIDANCE            GUIDANCE POLICY    3, EXECUTION ERROR CODE    1
       30.002          Q-LINEAR
      100.000          Q-LINEAR
      180.000          EIGENVALUE
      180.001          GUIDANCE            GUIDANCE POLICY    3, EXECUTION ERROR CODE    1
      180.002          Q-LINEAR
      190.000          PREDICTION          PREDICTING TO TIME 204.70
      200.000          PREDICTION          PREDICTING TO TIME 204.70
      203.000          PREDICTION          PREDICTING TO TIME 204.70
      206.680          EIGENVALUE

  FOR EIGENVALUE EVENTS, THE SIGMA LEVEL OF THE HYPERELLIPSOID IS K = 3

INITIAL COVARIANCE MATRIX
        1.00000000E+00
        0.                 1.00000000E+00
        0.                 0.                 1.00000000E+00
        0.                 0.                 0.                 9.00000000E-06
        0.                 0.                 0.                 0.                 9.00000000E-06
        0.                 0.                 0.                 0.                 0.                 9.00000000E-06
```

```
STATE TRANSITION MATRIX CODE . . . 1

DYNAMIC NOISE CONSTANTS
    1.000000000000E-22  1.000000000000E-22  1.000000000000E-22

MEASUREMENT NOISE IS CONSTANT
  RANGE (EARTH-CENTERED. . . . .  1.000000000000E-06
  RANGE-RATE (EARTH-CENTERED . .  1.000000000000E-12
  RANGE (STATION NUMBER 1) . . .  2.500000000000E-05
  RANGE-RATE (STATION NUMBER 1).  9.000000000000E-12
  RANGE (STATION NUMBER 2) . . .  2.500000000000E-05
  RANGE-RATE (STATION NUMBER 2).  9.000000000000E-12
  RANGE (STATION NUMBER 3) . . .  2.500000000000E-05
  RANGE-RATE (STATION NUMBER 3).  9.000000000000E-12
  STAR PLANET ANGLE NUMBER 1 . .  2.500000000000E-09
  STAR PLANET ANGLE NUMBER 2 . .  2.500000000000E-09
  STAR PLANET ANGLE NUMBER 3 . .  2.500000000000E-09
  APPARENT PLANET DIAMETER . . .  2.500000000000E-09

STATION LOCATION CONSTANTS
  STATION NO. 1     ALTITUDE =  1.03100000E+00     LATITUDE =  3.53840000E+01     LONGITUDE = -1.16833000E+02
  STATION NO. 2     ALTITUDE =  5.00000000E-02     LATITUDE =  4.04170000E+01     LONGITUDE = -3.66700000E+00
  STATION NO. 3     ALTITUDE =  5.00000000E-02     LATITUDE = -3.53110000E+01     LONGITUDE =  1.49136000E+02

ACTUAL DEVIATION OF STATE VECTOR AT INITIAL TIME
         5.00000000E-01
        -5.00000000E-01
         5.00000000E-01
        -1.50000000E-03
         1.50000000E-03
        -1.50000000E-03

ACTUAL TRAJECTORY INFORMATION

  BODIES TO BE CONSIDERED
    SUN
    EARTH
    MARS
    JUPITER
    MOON

ACCURACY FIGURE FOR ACTUAL TRAJECTORY. . 5.00000E-06

ACTUAL MEASUREMENT BIASES
  RANGE (EARTH-CENTERED. . . . .  0.
  RANGE-RATE (EARTH-CENTERED . .  0.
  RANGE (STATION NUMBER 1) . . .  0.
  RANGE-RATE (STATION NUMBER 1).  0.
  RANGE (STATION NUMBER 2) . . .  0.
  RANGE-RATE (STATION NUMBER 2).  0.
  RANGE (STATION NUMBER 3) . . .  0.
  RANGE-RATE (STATION NUMBER 3).  0.
  STAR PLANET ANGLE NUMBER 1 . .  0.
  STAR PLANET ANGLE NUMBER 2 . .  0.
  STAR PLANET ANGLE NUMBER 3 . .  0.
  APPARENT PLANET DIAMETER . . .  0.
```

```
DYNAMIC CONSTANT BIASES TO BE USED IN THE DETERMINATION OF THE ACTUAL TRAJECTORY
  GRAVITATIONAL CONSTANT OF SUN. . . . . . . 0.
  GRAVITATIONAL CONSTANT OF TARGET PLANET. . 0.
  SEMI-MAJOR AXIS OF TARGET PLANET . . . . . 0.
  ECCENTRICITY OF TARGET PLANET. . . . . . . 0.
  INCLINATION OF TARGET PLANET . . . . . . . 0.

ACTUAL UNMODELLED ACCELERATION TO BE USED TO CALCULATE THE ACTUAL DYNAMIC NOISE BY THE FOLLOWING SCHEDULE

                                                        X                         Y                         Z
  FROM   0.    DAYS THROUGH  210.607 DAYS. . .     0.                        0.                        0.

BIASES IN LOCATIONS OF ROTATING STATIONS

              ALTITUDE                  LATITUDE                  LONGITUDE
  1      0.                        0.                        0.
  2      0.                        0.                        0.
  3      0.                        0.                        0.

THE ACTUAL MEASUREMENT NOISE WILL BE CALCULATED FROM THE FOLLOWING CONSTANTS
  RANGE (EARTH-CENTERED. . . . . 0.
  RANGE-RATE (EARTH-CENTERED . . 0.
  RANGE (STATION NUMBER 1) . . . 2.5000000000000E-07
  RANGE-RATE (STATION NUMBER 1). 9.0000000000000E-14
  RANGE (STATION NUMBER 2) . . . 2.5000000000000E-07
  RANGE-RATE (STATION NUMBER 2). 9.0000000000000E-14
  RANGE (STATION NUMBER 3) . . . 2.5000000000000E-07
  RANGE-RATE (STATION NUMBER 3). 9.0000000000000E-14
  STAR PLANET ANGLE NUMBER 1 . . 0.
  STAR PLANET ANGLE NUMBER 2 . . 0.
  STAR PLANET ANGLE NUMBER 3 . . 0.
  APPARENT PLANET DIAMETER . . . 0.
```

SIMULATION MODE AT TRAJECTORY TIME .200 DAYS

INITIAL TRAJECTORY TIME 0. DAYS, CALENDAR DATE 7 24 9 25 47.637, 1973, JULIAN DATE 2441887.8929124624

FINAL TRAJECTORY TIME .20000 DAYS, CALENDAR DATE 7 24 14 13 47.636, 1973, JULIAN DATE 2441888.0929124504

STATE VECTOR AT TIME 0. DAYS

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	7.9030349297000E+07	7.9030349297000E+07	7.9030349797000E+07
RY	-1.2979967127000E+08	-1.2979967127000E+08	-1.2979967177000E+08
RZ	7.5601537160000E+01	7.5601537160000E+01	7.6101537160000E+01
VX	3.4316535000000E+01	3.4316535000000E+01	3.4315035000000E+01
VY	1.2503124000000E+01	1.2503124000000E+01	1.2504624000000E+01
VZ	6.2151790000000E+00	6.2151790000000E+00	6.2136790000000E+00

STATE VECTOR AT TIME .200 DAYS

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	7.9541152306374E+07	7.9541152306374E+07	7.9541086543283E+07
RY	-1.2948794292030E+08	-1.2948794292030E+08	-1.2948792212327E+08
RZ	4.4851189748346E+04	4.4851189748346E+04	4.4803522378908E+04
VX	2.8481883384022E+01	2.8481883384022E+01	2.8477183594286E+01
VY	1.8041138416805E+01	1.8041138416805E+01	1.8041820870270E+01
VZ	1.9356894727016E+00	1.9356894727016E+00	1.9324720828904E+00

SIMULATION MODE AT TRAJECTORY TIME .200 DAYS

AT INITIAL TIME, 0. DAYS

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
ORIGINAL NOMINAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	-1.13190969E+03	-6.57561453E+03	7.56015372E+01	6.67275367E+03
VELOCITY OF VEHICLE RELATIVE TO EARTH	9.35948271E+00	-2.88174107E+00	6.21517900E+00	1.15988274E+01
POSITION RELATIVE TO TARGET PLANET. .	-1.06945316E+08	-3.99079111E+07	6.44858919E+06	1.14330776E+08
VELOCITY RELATIVE TO TARGET PLANET. .	2.28283467E+01	-1.13737707E+01	5.99437813E+00	2.61997833E+01
MOST RECENT NOMINAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	-1.13190969E+03	-6.57561453E+03	7.56015372E+01	6.67275367E+03
VELOCITY OF VEHICLE RELATIVE TO EARTH	9.35948271E+00	-2.88174107E+00	6.21517900E+00	1.15988274E+01
POSITION RELATIVE TO TARGET PLANET. .	-1.06945316E+08	-3.99079111E+07	6.44858919E+06	1.14330776E+08
VELOCITY RELATIVE TO TARGET PLANET. .	2.28283467E+01	-1.13737707E+01	5.99437813E+00	2.61997833E+01
ACTUAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	-1.13140969E+03	-6.57611453E+03	7.61015372E+01	6.67316729E+03
VELOCITY OF VEHICLE RELATIVE TO EARTH	9.35798271E+00	-2.88024107E+00	6.21367900E+00	1.15964407E+01
POSITION RELATIVE TO TARGET PLANET. .	-1.06945316E+08	-3.99079116E+07	6.44858969E+06	1.14330776E+08
VELOCITY RELATIVE TO TARGET PLANET. .	2.28268467E+01	-1.13722707E+01	5.99287813E+00	2.61974820E+01

AT FINAL TIME .200 DAYS

	X-COMP.	Y-COMP.	Z-COMP.	RESULTANT
ORIGINAL NOMINAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	7.88420528E+04	3.85586720E+04	4.48511897E+04	9.85620094E+04
VELOCITY OF VEHICLE RELATIVE TO EARTH	3.57662457E+00	2.57153150E+00	1.93568947E+00	4.81164330E+00
POSITION RELATIVE TO TARGET PLANET. .	-1.06632608E+08	-4.00089690E+07	6.48953506E+06	1.14076048E+08
VELOCITY RELATIVE TO TARGET PLANET. .	1.70420417E+01	-5.85905870E+00	1.71321369E+00	1.81023439E+01
MOST RECENT NOMINAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	7.88420528E+04	3.85586720E+04	4.48511897E+04	9.85620094E+04
VELOCITY OF VEHICLE RELATIVE TO EARTH	3.57662457E+00	2.57153150E+00	1.93568947E+00	4.81164330E+00
POSITION RELATIVE TO TARGET PLANET. .	-1.06632608E+08	-4.00089690E+07	6.48953506E+06	1.14076048E+08
VELOCITY RELATIVE TO TARGET PLANET. .	1.70420417E+01	-5.85905870E+00	1.71321369E+00	1.81023439E+01
ACTUAL TRAJECTORY				
POSITION OF VEHICLE RELATIVE TO EARTH	7.87762897E+04	3.85794690E+04	4.48035224E+04	9.84956621E+04
VELOCITY OF VEHICLE RELATIVE TO EARTH	3.57192478E+00	2.57221395E+00	1.93247208E+00	4.80722161E+00
POSITION RELATIVE TO TARGET PLANET. .	-1.06632674E+08	-4.00089482E+07	6.48948739E+06	1.14076099E+08
VELOCITY RELATIVE TO TARGET PLANET. .	1.70373419E+01	-5.85837625E+00	1.70999630E+00	1.80973943E+01

STATISTICAL DATA AFTER MEASUREMENT 1

RANGE AND RANGE-RATE WERE MEASURED FROM STATION 1 AT TRAJECTORY TIME .20000 DAYS

STATE TRANSITION MATRIX -- PSI(.200, 0.)

-3.84052651E+00	-2.54341816E+01	6.32241324E+00	2.67218447E+04	-1.27251051E+04	1.36854477E+04
-5.33885777E+00	1.25624341E+01	-5.93403792E+00	-3.73488032E+03	1.48456059E+04	-3.24092384E+03
3.92155873E+00	-1.85391569E+01	-4.40068115E+00	1.26113560E+04	-8.88018869E+03	1.62995716E+04
-2.80562749E-04	-1.80537441E-03	3.72341250E-04	1.84202876E+00	-8.63030537E-01	1.05908398E+00
-2.90673906E-04	3.97387941E-04	-2.93163010E-04	-5.63808286E-02	6.51762048E-01	-7.52420204E-02
1.91374993E-04	-1.24328356E-03	-2.61710514E-04	9.62710886E-01	-5.81226182E-01	9.91312254E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

2.22902511E-06	2.22902511E-06	2.22902511E-06	2.98598400E-14	2.98598400E-14	2.98598400E-14

OBSERVATION MATRIX

7.66955407E-01	3.88568404E-01	5.10679939E-01	0.	0.	0.
5.40348798E-07	4.66422432E-06	-4.36044861E-06	7.66955407E-01	3.88568404E-01	5.10679939E-01

MEASUREMENT NOISE MATRIX

2.50000000E-05	0.
0.	9.00000000E-12

K MATRIX

-5.03527666E+00	8.52240967E+04
-8.78785092E-01	8.40195461E+03
1.01889531E+01	-1.34384992E+05
-2.16765312E-04	4.06173478E+00
-5.77516742E-05	6.95450443E-01
4.69908036E-04	-5.98634838E+00

SIMULATION MODE AT TRAJECTORY TIME .200 DAYS

PROBLEM. • 402 PAGE. • 4

COVARIANCE MATRIX AT TIME .20000 DAYS, JUST BEFORE THE MEASUREMENT

1.02711110E+04	-3.33414316E+03	6.48624719E+03	7.21637218E-01	-1.08314943E-01	4.49425421E-01
-3.33414316E+03	2.42513534E+03	-2.31355398E+03	-2.31510908E-01	9.94557528E-02	-1.54020682E-01
6.48624719E+03	-2.31355398E+03	4.91066572E+03	4.64144048E-01	-7.67440075E-02	3.26095698E-01
7.21637218E-01	-2.31510908E-01	4.64144048E-01	5.08126853E-05	-7.45933792E-06	3.20170142E-05
-1.08314943E-01	9.94557528E-02	-7.67440075E-02	-7.45933792E-06	4.23105841E-06	-5.04216124E-06
4.49425421E-01	-1.54020682E-01	3.26095698E-01	3.20170142E-05	-5.04216124E-06	2.18768958E-05

COVARIANCE MATRIX AT TIME .20000 DAYS, AFTER CONSIDERING THE MEASUREMENT

2.13031513E+02	-5.42395174E+02	9.27622702E+01	1.25945461E-02	-2.55267692E-02	6.03006718E-03
-5.42395174E+02	1.63113359E+03	-4.26517804E+02	-3.35465555E-02	7.56438086E-02	-2.51404150E-02
9.27622702E+01	-4.26517804E+02	1.85217913E+02	6.61014983E-03	-1.92192821E-02	1.00728258E-02
1.25945461E-02	-3.35465555E-02	6.61014983E-03	7.58383893E-07	-1.57262714E-06	4.07201382E-07
-2.55267692E-02	7.56438086E-02	-1.92192821E-02	-1.57262714E-06	3.51332731E-06	-1.13937873E-06
6.03006718E-03	-2.51404150E-02	1.00728258E-02	4.07201382E-07	-1.13937873E-06	5.64525079E-07

ACTUAL DYNAMIC NOISE

0.
0.
0.
0.
0.
0.

MATRIX OF VARIANCES OF ACTUAL MEASUREMENT NOISE

2.50000000E-07	0.
0.	9.00000000E-14

ACTUAL MEASUREMENT NOISE

-3.82242800E-04
2.97966270E-07

MEASUREMENT

ESTIMATED	ACTUAL	RESIDUAL
9.81819552E+04	9.81152690E+04	-6.66862867E+01
4.82961520E+00	4.82490342E+00	-4.71178246E-03

RESIDUAL UNCERTAINTIES

9.86400891E+03	6.98890970E-01
6.98890970E-01	4.95366188E-05

DEVIATION OF THE STATE VECTOR FROM MOST RECENT NOMINAL

ESTIMATED	ACTUAL
-6.57735003E+01	-6.57630901E+01
1.90147323E+01	2.07970357E+01
-4.62706011E+01	-4.76673694E+01
-4.68273696E-03	-4.69978974E-03
5.74433503E-04	6.82453465E-04
-3.13005069E-03	-3.21738981E-03

DEVIATION FROM ORIGINAL NOMINAL

ESTIMATED	ACTUAL
-6.57735003E+01	-6.57630901E+01
1.90147323E+01	2.07970357E+01
-4.62706011E+01	-4.76673694E+01
-4.68273696E-03	-4.69978974E-03
5.74433503E-04	6.82453465E-04
-3.13005069E-03	-3.21738981E-03

SIMULATION MODE AT TRAJECTORY TIME .200 DAYS

ACTUAL ORBIT DETERMINATION INACCURACY

```
 1.04101682E-02
 1.78230344E+00
-1.39676835E+00
-1.70527780E-05
 1.08019962E-04
-8.73391250E-05
```

****

STATE VECTOR

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	8.5035623158E+07	8.5035623158E+07	8.5034471684E+07
RY	-1.2586682556E+08	-1.2586682556E+08	-1.2586684509E+08
RZ	3.7063954958E+05	3.7063954958E+05	3.6988344359E+05
VX	2.7220991707E+01	2.7220991707E+01	2.7215317746E+01
VY	1.8626079102E+01	1.8626079102E+01	1.8625629604E+01
VZ	1.5746287990E+00	1.5746287990E+00	1.5709685203E+00

STATE TRANSITION MATRIX RELATING THE STATE VECTOR AT TIME 2.500 DAYS TO THAT AT TIME 2.200 DAYS

1.0001456077E+00	2.5942086663E-04	1.9734316088E-04	2.5921182264E+04	2.1102210308E+00	1.6033600273E+00
2.5942093049E-04	9.9996614308E-01	1.4055503099E-04	2.1102212684E+00	2.5919726950E+04	1.1431271019E+00
1.9734315261E-04	1.4055499050E-04	9.9988829476E-01	1.6033599965E+00	1.1431269512E+00	2.5919091006E+04
1.0591973075E-08	1.8899763319E-08	1.4361740926E-08	1.0001288986E+00	2.3044432277E-04	1.7490041490E-04
1.8899774696E-08	-2.4459680130E-09	1.0238350902E-08	2.3044437717E-04	9.9997043136E-01	1.2481388715E-04
1.4361739451E-08	1.0238343687E-08	-8.1394583773E-09	1.7490040784E-04	1.2481385265E-04	9.9990070945E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

1.1284439630E-05	1.1284439630E-05	1.1284439630E-05	6.7184640000E-14	6.7184640000E-14	6.7184640000E-14

COVARIANCE MATRIX AT TIME OF QUASI-LINEAR FILTERING EVENT -- P(2.500, 2.200)

9.8624442471E-01	-8.4102058361E-01	-6.9267942684E-01	6.6149617404E-06	-5.1913681467E-06	-5.0859886944E-06
-8.4102058361E-01	9.5241611590E-01	2.8666659880E-01	-5.2404919990E-06	5.2990937133E-06	2.4772542878E-06
-6.9267942684E-01	2.8666659880E-01	8.8063981429E-01	-5.1621011120E-06	2.5318547674E-06	5.9939327599E-06
6.6149617404E-06	-5.2404919990E-06	-5.1621011120E-06	4.8208531834E-11	-3.5629257266E-11	-3.9803857782E-11
-5.1913681467E-06	5.2990937133E-06	2.5318547674E-06	-3.5629257266E-11	3.2745966249E-11	2.1730284021E-11
-5.0859886944E-06	2.4772542878E-06	5.9939327599E-06	-3.9803857782E-11	2.1730284021E-11	4.3765803919E-11

CORRELATION COEFFICIENT MATRIX AT TIME 2.500 DAYS

1.0000000000E+00	-8.6776269666E-01	-7.4326015262E-01	9.5934121594E-01	-9.1350439792E-01	-7.7413313126E-01
-8.6776269666E-01	1.0000000000E+00	3.1301458047E-01	-7.7338677578E-01	9.4887570540E-01	3.8369820357E-01
-7.4326015262E-01	3.1301458047E-01	1.0000000000E+00	-7.9225574311E-01	4.7147723390E-01	9.6548438319E-01
9.5934121594E-01	-7.7338677578E-01	-7.9225574311E-01	1.0000000000E+00	-8.9673869723E-01	-8.6655425276E-01
-9.1350439792E-01	9.4887570540E-01	4.7147723390E-01	-8.9673869723E-01	1.0000000000E+00	5.7400945044E-01
-7.7413313126E-01	3.8369820357E-01	9.6548438319E-01	-8.6655425276E-01	5.7400945044E-01	1.0000000000E+00

****

ACTUAL DYNAMIC NOISE

```
0.
0.
0.
0.
0.
0.
```

DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NOMINAL TRAJECTORY

ESTIMATED	ACTUAL
-1.1467405258E+03	-1.1514733119E+03
-2.3322947393E+01	-1.9528353214E+01
-7.5898892756E+02	-7.5610598306E+02
-5.6296463792E-03	-5.6739616766E-03
-4.8527517745E-04	-4.4949866253E-04
-3.6884891386E-03	-3.6602787035E-03

STATE VECTOR OF NEW NOMINAL TRAJECTORY

```
 8.5034476417E+07
-1.2586684889E+08
 3.6988056065E+05
 2.7215362061E+01
 1.8625593827E+01
 1.5709403098E+00
```

ACTUAL DEVIATION OF NEW STATE VECTOR

```
-4.7327861099E+00
 3.7945941789E+00
 2.8829445068E+00
-4.4315297449E-05
 3.5776514915E-05
 2.8210435062E-05
```

****

STATE VECTOR

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	8.6208254123E+07	8.6206863532E+07	8.6206856886E+07
RY	-1.2505795114E+08	-1.2505799617E+08	-1.2505799083E+08
RZ	4.3848866282E+05	4.3757006652E+05	4.3757416649E+05
VX	2.7067913173E+01	2.7062254995E+01	2.7062210752E+01
VY	1.8822102218E+01	1.8821584525E+01	1.8821620232E+01
VZ	1.5669244403E+00	1.5632244426E+00	1.5632525776E+00

STATE TRANSITION MATRIX -- PSI(3.000, 3.000)

1.00000000E+00	3.96196589E-10	-1.38977718E-12	-2.94266158E-03	-1.57484390E-03	-1.30790274E-04
-1.90013602E-10	1.00000000E+00	-9.66449143E-13	-1.57453231E-03	-1.77367251E-03	-9.09536085E-05
-1.57809321E-11	2.28861374E-11	1.00000000E+00	-1.30782624E-04	-9.09662868E-05	-6.86058612E-04
3.38124806E-17	-1.17687832E-17	4.13443283E-20	1.00000000E+00	1.89963344E-10	1.57753810E-11
-1.17888462E-17	4.27888947E-17	-6.00499476E-20	-3.96250401E-10	1.00000000E+00	-2.28901342E-11
4.08517842E-20	-5.92335912E-20	2.56959716E-17	1.38355993E-12	9.62674385E-13	1.00000000E+00

DIAGONAL OF DYNAMIC NOISE MATRIX

0.	0.	0.	0.	0.	0.

COVARIANCE MATRIX AT TIME OF PREDICTION EVENT -- P(3.000, 3.000)

8.06306790E-01	-6.02590353E-01	-7.01402350E-01	4.31183248E-06	-3.20032946E-06	-3.93759134E-06
-6.02590353E-01	6.51506356E-01	2.51695158E-01	-2.98448907E-06	3.00367883E-06	1.57648198E-06
-7.01402350E-01	2.51695158E-01	9.79363308E-01	-4.07213325E-06	1.95562882E-06	5.27673458E-06
4.31183248E-06	-2.98448907E-06	-4.07213325E-06	2.52512294E-11	-1.78115383E-11	-2.41622039E-11
-3.20032946E-06	3.00367883E-06	1.95562882E-06	-1.78115383E-11	1.58950353E-11	1.24182656E-11
-3.93759134E-06	1.57648198E-06	5.27673458E-06	-2.41622039E-11	1.24182656E-11	3.20772952E-11

```
SIMULATION MODE     --  PREDICTION EVENT AT TRAJECTORY TIME        3.000 DAYS, PREDICTING TO TRAJECTORY TIME    5.000 DAYS
                                                                             PROBLEM. .       402     PAGE. .       64

**************************************************************************************************************************


          POSITION EIGENVALUES
             1     1.4803458749E-05
             2     5.4748959122E-01
             3     1.8896720602E+00


          POSITION EIGENVECTORS
             1     7.4531873630E-01    -1.5836358651E-01    -6.4762717345E-01
             2     5.3641144249E-01     7.1930301353E-01     4.4143622313E-01
             3     3.9593275399E-01    -6.7640531429E-01     6.2105805294E-01

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                       THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  3.753E+04 X**2 +  1.944E+04 Y**2 +  1.059E+04 Z**2 +  5.401E+04 XY +  3.987E+04 XZ +  2.869E+04 YZ =   9

          XY HYPERELLIPSOID. . . . . .  3.753E+04 X**2 +  5.401E+04 XY +  1.944E+04 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  3.753E+04 X**2 +  3.987E+04 XZ +  1.059E+04 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  1.944E+04 Y**2 +  2.869E+04 YZ +  1.059E+04 Z**2 =   9


          VELOCITY EIGENVALUES
             1     7.7942471258E-14
             2     9.3562344314E-12
             3     6.1789382966E-11


          VELOCITY EIGENVECTORS
             1     7.4791855180E-01    -1.9617521142E-01    -6.3413967413E-01
             2     5.4707996857E-01     7.2321103605E-01     4.2150836924E-01
             3     3.7592731727E-01    -6.6217904208E-01     6.4822647922E-01
```

```
*******************************************************************************************************

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                     THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

 7.187E+12 X**2 + 3.899E+12 Y**2 + 1.867E+12 Z**2 + 1.046E+13 XY + 7.229E+12 XZ + 5.184E+12 YZ =  9

      XY HYPERELLIPSOID. . . . . .  7.187E+12 X**2 +  1.046E+13 XY +  3.899E+12 Y**2 =   9

      XZ HYPERELLIPSOID. . . . . .  7.187E+12 X**2 +  7.229E+12 XZ +  1.867E+12 Z**2 =   9

      YZ HYPERELLIPSOID. . . . . .  3.899E+12 Y**2 +  5.184E+12 YZ +  1.867E+12 Z**2 =   9


CORRELATION COEFFICEINT MATRIX AT TIME OF PREDICTION EVENT    3.000 DAYS

       1.00000000E+00  -8.31405107E-01  -7.89305526E-01   9.55587495E-01  -8.93951140E-01  -7.99578473E-01
      -8.31405107E-01   1.00000000E+00   3.15096528E-01  -7.35816293E-01   9.33390588E-01   3.56131421E-01
      -7.89305526E-01   3.15096528E-01   1.00000000E+00  -8.18858390E-01   4.95659879E-01   9.72241032E-01
       9.55587495E-01  -7.35816293E-01  -8.18858390E-01   1.00000000E+00  -8.89056613E-01  -8.76750237E-01
      -8.93951140E-01   9.33390588E-01   4.95659879E-01  -8.89056613E-01   1.00000000E+00   5.67950919E-01
      -7.99578473E-01   3.56131421E-01   9.72241032E-01  -8.76750237E-01   5.67950919E-01   1.00000000E+00


ACTUAL DYNAMIC NOISE

   0.
   0.
   0.
   0.
   0.
   0.


DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NOMINAL TRAJECTORY

      ESTIMATED             ACTUAL
  -4.6934948367E+00   -6.6455903053E+00
  -1.7964440197E+00    5.3385534286E+00
   1.0041701765E+01    4.0999683030E+00
  -2.9568274430E-05   -4.4243259822E-05
  -2.1963159371E-06    3.5707065990E-05
   4.8594795042E-05    2.8135047977E-05
```

SIMULATION MODE -- PREDICTION EVENT AT TRAJECTORY TIME 3.000 DAYS, PREDICTING TO TRAJECTORY TIME 5.000 DAYS

PROBLEM. . 402 PAGE. . 66

****

STATE TRANSITION MATRIX -- PSI(5.000, 3.000)

1.00000103E+00	-7.99937293E-04	3.40972531E-06	1.72800636E+05	-4.62742495E+01	2.15059674E-01
-7.99929668E-04	1.00056467E+00	-4.80552686E-06	-4.62739664E+01	1.72831946E+05	-2.99031839E-01
3.40991250E-06	-4.80583711E-06	9.99434615E-01	2.15066624E-01	-2.99043358E-01	1.72767433E+05
1.29386204E-10	-9.29882721E-09	4.32313095E-11	1.00002111E+00	-8.06738284E-04	4.06020496E-06
-9.29861909E-09	6.42119832E-09	-6.00766787E-11	-8.06730659E-04	1.00054459E+00	-5.57516050E-06
4.32364187E-11	-6.00851469E-11	-6.54318740E-09	4.06039215E-06	-5.57547075E-06	9.99434616E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

2.22902511E-02	2.22902511E-02	2.22902511E-02	2.98598400E-12	2.98598400E-12	2.98598400E-12

COVARIANCE MATRIX AT PREDICTION TIME -- P(5.000, 3.000)

3.07513604E+00	-2.20684780E+00	-2.80649606E+00	8.69422079E-06	-6.31321589E-06	-8.10066286E-06
-2.20684780E+00	2.19058468E+00	1.23428517E+00	-6.08494341E-06	5.78217121E-06	3.72172963E-06
-2.80649606E+00	1.23428517E+00	3.72055562E+00	-8.25264096E-06	4.12488099E-06	1.04516616E-05
8.69422079E-06	-6.08494341E-06	-8.25264096E-06	2.83232053E-11	-1.79420865E-11	-2.41466358E-11
-6.31321589E-06	5.78217121E-06	4.12488099E-06	-1.79420865E-11	1.90251404E-11	1.24703519E-11
-8.10066286E-06	3.72172963E-06	1.04516616E-05	-2.41466358E-11	1.24703519E-11	3.59594429E-11

POSITION EIGENVALUES

1	2.4797222904E-02
2	1.6099293332E+00
3	7.3515497786E+00

POSITION EIGENVECTORS

1	7.4700708194E-01	-1.8092354448E-01	-6.3972423011E-01
2	5.4081979101E-01	7.2500636220E-01	4.2647359640E-01
3	3.8664502220E-01	-6.6455432120E-01	6.3943192053E-01

```
************************************************************************************************************************

              FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                    THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

2.258E+01 X**2 + 1.215E+01 Y**2 + 6.359E+00 Z**2 + 3.235E+01 XY + 2.333E+01 XZ + 1.634E+01 YZ = 9

      XY HYPERELLIPSOID. . . . . .  2.258E+01 X**2 + 3.235E+01 XY + 1.215E+01 Y**2 =  9

      XZ HYPERELLIPSOID. . . . . .  2.258E+01 X**2 + 2.333E+01 XZ + 6.359E+00 Z**2 =  9

      YZ HYPERELLIPSOID. . . . . .  1.215E+01 Y**2 + 1.634E+01 YZ + 6.359E+00 Z**2 =  9


      VELOCITY EIGENVALUES
        1    3.0638349693E-12
        2    1.2365931240E-11
        3    6.4878022317E-11


      VELOCITY EIGENVECTORS
        1    7.4793908417E-01   -1.9394206805E-01   -6.3480201686E-01
        2    5.4706132825E-01    7.2173171341E-01    4.2406041666E-01
        3    3.7591359312E-01   -6.6444699419E-01    6.4590956210E-01

              FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                    THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

1.918E+11 X**2 + 1.426E+11 Y**2 + 8.825E+10 Z**2 + 2.362E+11 XY + 1.917E+11 XZ + 6.513E+10 YZ = 9

      XY HYPERELLIPSOID. . . . . .  1.918E+11 X**2 + 2.362E+11 XY + 1.426E+11 Y**2 =  9

      XZ HYPERELLIPSOID. . . . . .  1.918E+11 X**2 + 1.917E+11 XZ + 8.825E+10 Z**2 =  9

      YZ HYPERELLIPSOID. . . . . .  1.426E+11 Y**2 + 6.513E+10 YZ + 8.825E+10 Z**2 =  9
```

```
SIMULATION MODE     --  PREDICTION EVENT AT TRAJECTORY TIME        3.000 DAYS, PREDICTING TO TRAJECTORY TIME    5.000 DAYS
                                                                                PROBLEM. .       402     PAGE. .      68

******************************************************************************************************************************

CORRELATION COEFFICEINT MATRIX AT TIME OF PREDICTION EVENT    5.000 DAYS

          1.00000000E+00  -8.50276533E-01  -8.29713909E-01   9.31595320E-01  -8.25381136E-01  -8.04634149E-01
         -8.50276533E-01   1.00000000E+00   4.32346142E-01  -7.72512154E-01   8.95667856E-01   4.38000763E-01
         -8.29713909E-01   4.32346142E-01   1.00000000E+00  -8.03929288E-01   4.90279333E-01   9.43825070E-01
          9.31595320E-01  -7.72512154E-01  -8.03929288E-01   1.00000000E+00  -7.72925812E-01  -7.90306402E-01
         -8.25381136E-01   8.95667856E-01   4.90279333E-01  -7.72925812E-01   1.00000000E+00   4.97994838E-01
         -8.04634149E-01   4.38000763E-01   9.43825070E-01  -7.90306402E-01   4.97994838E-01   1.00000000E+00

COVARIANCE OF UNCERTAINTIES IN B DOT T AND B DOT R AT SPHERE OF INFLUENCE

        0.                         0.
        0.                         0.


          EIGENVALUES OF ABOVE MATRIX

             1     0.
             2     0.


          EIGENVECTORS OF ABOVE MATRIX

             1     1.0000000000E+00     0.
             2     0.                   1.0000000000E+00
```

SIMULATION MODE --EIGENVECTOR EVENT AT TRAJECTORY TIME 5.000 DAYS PROBLEM. . 402 PAGE. . 105

****

STATE VECTOR

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	9.0833275299E+07	9.0830900906E+07	9.0830886630E+07
RY	-1.2173732976E+08	-1.2173747047E+08	-1.2173745898E+08
RZ	7.0750098471E+05	7.0594092371E+05	7.0594986482E+05
VX	2.6461725590E+01	2.6456006446E+01	2.6455962343E+01
VY	1.9610438737E+01	1.9609862021E+01	1.9609897648E+01
VZ	1.5486997347E+00	1.5449802609E+00	1.5450081732E+00

STATE TRANSITION MATRIX -- PSI(5.000, 4.900)

1.00000010E+00	-2.03130921E-06	1.16200143E-08	8.64000224E+03	-4.75750059E-03	1.20075027E-04
-2.03090804E-06	1.00000132E+00	-1.56115604E-08	-4.74665685E-03	8.64000506E+03	1.92512342E-05
1.16297747E-08	-1.56276995E-08	9.99998586E-01	1.20338846E-04	1.88233671E-05	8.63999640E+03
2.29474206E-11	-4.70236896E-10	2.69919934E-12	1.00000010E+00	-2.03193838E-06	1.16912391E-08
-4.70236895E-10	3.04254038E-10	-3.62469800E-12	-2.03153720E-06	1.00000131E+00	-1.56897286E-08
2.69919937E-12	-3.62469804E-12	-3.27200604E-10	1.17009993E-08	-1.57059284E-08	9.99998586E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

1.39314070E-07	1.39314070E-07	1.39314070E-07	7.46496000E-15	7.46496000E-15	7.46496000E-15

COVARIANCE MATRIX AT TIME OF EIGENVALUE EVENT -- P(5.000, 4.900)

1.07819732E+00	-1.02152730E+00	-6.24311495E-01	2.97194980E-06	-2.84517651E-06	-1.98461158E-06
-1.02152730E+00	1.19839099E+00	2.76266916E-01	-2.70717511E-06	3.18162865E-06	9.94039656E-07
-6.24311495E-01	2.76266916E-01	7.92758439E-01	-1.86655898E-06	9.85008267E-07	2.36302023E-06
2.97194980E-06	-2.70717511E-06	-1.86655898E-06	8.93124356E-12	-8.08868470E-12	-6.30010696E-12
-2.84517651E-06	3.18162865E-06	9.85008267E-07	-8.08868470E-12	9.17287289E-12	3.53922980E-12
-1.98461158E-06	9.94039656E-07	2.36302023E-06	-6.30010696E-12	3.53922980E-12	7.73493277E-12

```
SIMULATION MODE     --EIGENVECTOR EVENT AT TRAJECTORY TIME        5.000 DAYS
                                                                          PROBLEM. .       402     PAGE. .      106

*****************************************************************************************************************

          POSITION EIGENVALUES
             1       4.0180275706641E-05
             2       2.4110115121754E+00
             3       6.5829504928569E-01


          POSITION EIGENVECTORS
             1       7.4194377129814E-01      -6.6806603031560E-01      -5.6632317365636E-02
             2       5.4124186674264E-01       6.4665778091112E-01      -5.3748577290194E-01
             3       3.9569771532924E-01       3.6813244019704E-01       8.4136901806387E-01

                     FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                             THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

   1.370E+04 X**2 .  7.291E+03 Y**2 +  3.898E+03 Z**2 +  1.999E+04 XY +  1.461E+04 XZ +  1.066E+04 YZ =    9

          XY HYPERELLIPSOID. . . . . .  1.370E+04 X**2 +  1.999E+04 XY +  7.291E+03 Y**2 =    9

          XZ HYPERELLIPSOID. . . . . .  1.370E+04 X**2 .  1.461E+04 XZ +  3.898E+03 Z**2 =    9

          YZ HYPERELLIPSOID. . . . . .  7.291E+03 Y**2 +  1.066E+04 YZ +  3.898E+03 Z**2 =    9


          VELOCITY EIGENVALUES
             1       1.2228015132778E-13
             2       2.0868860987835E-11
             3       4.8479080734258E-12


          VELOCITY EIGENVECTORS
             1       7.5828427190361E-01      -6.5152005087677E-01       2.2947468032082E-02
             2       5.2834832205810E-01       5.9354609199006E-01      -6.0708408582473E-01
             3       3.8190707451152E-01       4.7246657023411E-01       7.9430619187391E-01
```

****

FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

4.723E+12 X**2 + 2.376E•12 Y**2 • 1.334E+12 Z**2 + 6.510E•12 XY • 4.715E+12 XZ + 3.128E+12 YZ = 9

XY HYPERELLIPSOID. . . . . . . 4.723E+12 X**2 + 6.510E•12 XY + 2.376E•12 Y**2 = 9

XZ HYPERELLIPSOID. . . . . . . 4.723E+12 X**2 • 4.715E•12 XZ + 1.334E+12 Z**2 = 9

YZ HYPERELLIPSOID. . . . . . . 2.376E+12 Y**2 • 3.128E•12 YZ • 1.334E+12 Z**2 = 9

CORRELATION COEFFECIENT MATRIX

1.00000000E+00	-8.98672776E-01	-6.75276709E-01	9.57715198E-01	-9.04705861E-01	-6.87223698E-01
-8.98672776E-01	1.00000000E+00	2.83438718E-01	-8.27486878E-01	9.59616120E-01	3.26494727E-01
-6.75276709E-01	2.83438718E-01	1.00000000E+00	-7.01480035E-01	3.65272310E-01	9.54264418E-01
9.57715198E-01	-8.27486878E-01	-7.01480035E-01	1.00000000E+00	-8.93653689E-01	-7.57990364E-01
-9.04705861E-01	9.59616120E-01	3.65272310E-01	-8.93653689E-01	1.00000000E+00	4.20172672E-01
-6.87223698E-01	3.26494727E-01	9.54264418E-01	-7.57990364E-01	4.20172672E-01	1.00000000E+00

ACTUAL DYNAMIC NOISE

0.
0.
0.
0.
0.
0.

SIMULATION MODE --EIGENVECTOR EVENT AT TRAJECTORY TIME 5.000 DAYS PROBLEM. . 402 PAGE. . 108

****

DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NOMINAL TRAJECTORY

ESTIMATED	ACTUAL
-1.1357918296E+01	-1.4275577068E+01
3.8874899913E+00	1.1497767925E+01
1.3880507724E+01	8.9411029518E+00
-3.1788922197E-05	-4.4102898414E-05
1.2079630642E-05	3.5626402791E-05
3.6820621426E-05	2.7912268223E-05

****

STATE VECTOR

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	9.0835561579E+07	9.0833186691E+07	9.0833172412E+07
RY	-1.2173563540E+08	-1.2173577617E+08	-1.2173576467E+08
RZ	7.0763479207E+05	7.0607440972E+05	7.0608335323E+05
VX	2.6461420160E+01	2.6455700097E+01	2.6455656894E+01
VY	1.9610831065E+01	1.9610254336E+01	1.9610289962E+01
VZ	1.5486929861E+00	1.5449735087E+00	1.5450014209E+00

STATE TRANSITION MATRIX -- PSI(5.001, 5.000)

1.0000000002E+00	-4.5930443476E-10	2.6674218390E-12	8.6401931799E+01	1.0893566969E-03	8.5818005971E-05
-6.1601237999E-11	9.9999999994E-01	-4.7628567756E-13	1.0889883816E-03	8.6401269646E+01	6.3598148770E-05
1.2340350963E-11	-1.6531886970E-11	9.9999999986E-01	8.5809047818E-05	6.3613017645E-05	8.6400467336E+01
2.3834254412E-13	-4.7049688698E-12	2.7286316840E-14	9.9999999982E-01	-3.4495679230E-10	-9.9822372590E-12
-4.7049688692E-12	3.0335787107E-12	-3.6570178730E-14	5.2746404466E-11	1.0000000003E+00	1.3372192242E-11
2.7286316854E-14	-3.6570178754E-14	-3.2719907270E-12	-3.0908609006E-13	-2.6831870059E-12	9.9999999986E-01

DIAGONAL OF DYNAMIC NOISE MATRIX

1.3931406951E-15	1.3931406951E-15	1.3931406951E-15	7.4649600001E-19	7.4649600001E-19	7.4649600001E-19

COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT -- P(5.001, 5.000)

1.0787109435E+00	-1.0220070820E+00	-6.2464428639E-01	2.9727265173E-06	-2.8458835360E-06	-1.9851538116E-06
-1.0220070820E+00	1.1989408415E+00	2.7643793165E-01	-2.7078798510E-06	3.1824296260E-06	9.9434446715E-07
-6.2464428639E-01	2.7643793165E-01	7.9316682822E-01	-1.8671047420E-06	9.8531780575E-07	2.3636859043E-06
2.9727265173E-06	-2.7078798510E-06	-1.8671047420E-06	8.9312710958E-12	-8.0887224469E-12	-6.3001057527E-12
-2.8458835360E-06	3.1824296260E-06	9.8531780575E-07	-8.0887224469E-12	9.1729196439E-12	3.5392386494E-12
-1.9851538116E-06	9.9434446715E-07	2.3636859043E-06	-6.3001057527E-12	3.5392386494E-12	7.7349178668E-12

```
SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME    5.001 DAYS
                                                                                PROBLEM. .        402      PAGE. .      110

*********************************************************************************************************************************

          POSITION EIGENVALUES OF ABOVE MATRIX
             1    4.0452028200E-05
             2    2.4121856875E+00
             3    6.5859247375E-01


          POSITION EIGENVECTORS OF ABOVE MATRIX
             1    7.4194822326E-01   -2.6806296768E-01   -5.6610115843E-02
             2    5.4123906871E-01    6.4664493138E-01   -5.3750404950E-01
             3    3.9569319493E-01    3.6816056814E-01    8.4135883638E-01

                FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                       THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

 1.361E+04 X**2 + 7.242E+03 Y**2 + 3.872E+03 Z**2 + 1.985E+04 XY + 1.451E+04 XZ + 1.059E+04 YZ =   9

          XY HYPERELLIPSOID. . . . . .  1.361E+04 X**2 + 1.985E+04 XY + 7.242E+03 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  1.361E+04 X**2 + 1.451E+04 XZ + 3.872E+03 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  7.242E+03 Y**2 + 1.059E+04 YZ + 3.872E+03 Z**2 =   9


          VELOCITY EIGENVALUES OF ABOVE MATRIX
             1    1.2228088295E-13
             2    2.0868919244E-11
             3    4.8479084794E-12


          VELOCITY EIGENVECTORS OF ABOVE MATRIX
             1    7.5828423035E-01   -6.5152015104E-01    2.2945997258E-02
             2    5.2834839468E-01    5.9354747177E-01   -6.0708267360E-01
             3    3.8190705654E-01    4.7246469873E-01    7.9430731371E-01
```

```
************************************************************************************************************

              FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                    THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

 4.723E+12 X**2 +  2.376E+12 Y**2 +  1.334E+12 Z**2 +  6.510E+12 XY +  4.715E+12 XZ +  3.128E+12 YZ =   9

       XY HYPERELLIPSOID. . . . . .  4.723E+12 X**2 +  6.510E+12 XY +  2.376E+12 Y**2 =   9

       XZ HYPERELLIPSOID. . . . . .  4.723E+12 X**2 +  4.715E+12 XZ +  1.334E+12 Z**2 =   9

       YZ HYPERELLIPSOID. . . . . .  2.376E+12 Y**2 +  3.128E+12 YZ +  1.334E+12 Z**2 =   9


ACTUAL DYNAMIC NOISE

   0.
   0.
   0.
   0.
   0.
   0.


DEVIATION IN THE STATE VECTOR FROM THE MOST RECENT NOMINAL TRAJECTORY

      ESTIMATED              ACTUAL
 -1.1360664908E+01    -1.4279389381E+01
  3.8885336549E+00     1.1500848770E+01
  1.3883689039E+01     8.9435145967E+00
 -3.1788942815E-05    -4.4102878405E-05
  1.2079695369E-05     3.5626422800E-05
  3.6820575552E-05     2.7912176009E-05
```

SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME 5.001 DAYS PROBLEM. . 402 PAGE. .

****

STATE TRANSITION MATRIX -- PSI(5.001, 0.)

-1.4160811275E+02	-9.0416311705E+02	1.4266847802E+02	9.3287903780E+05	-3.9944894950E+05	5.5968592823E+05
-1.3009802239E+02	3.6619061994E+00	-1.0556073176E+02	1.3019701195E+05	1.9226162045E+05	7.2640436400E+04
5.0188372267E+01	-5.9110566439E+02	-9.4737238334E+01	4.9640738116E+05	-2.5958861571E+05	4.5909215067E+05
-3.4223281947E-04	-2.1746840954E-03	3.2181165106E-04	2.2458525876E+00	-9.4497687614E-01	1.3608094653E+00
-3.0558581210E-04	-9.7306581629E-05	-2.3275379480E-04	3.9494361771E-01	3.9107541335E-01	2.3164594802E-01
9.8504324458E-05	-1.4051761225E-03	-2.1139695443E-04	1.2007247518E+00	-6.0884099605E-01	1.0823248425E+00

DIAGONAL OF DYNAMIC NOISE MATRIX

8.7140971375E-01	8.7140971375E-01	8.7140971375E-01	1.8669865706E-11	1.8669865706E-11	1.8669865706E-11

COVARIANCE MATRIX RELATING THE TIME OF THIS GUIDANCE EVENT TO THAT AT THE LAST GUIDANCE EVENT -- P(5.001, 0.)

1.2945558872E+07	7.6788850489E+05	7.9273835496E+06	3.1168487044E+01	3.1748682509E+00	1.8948244213E+01
7.6788850489E+05	5.6081467506E+05	4.3394090677E+05	1.8887223875E+00	1.3548946524E+00	1.0654054815E+00
7.9273835496E+06	4.3394090677E+05	5.0820500575E+06	1.9101904294E+01	1.8721614131E+00	1.2114431446E+01
3.1168487044E+01	1.8887223875E+00	1.9101904294E+01	7.5047693509E-05	7.7351749847E-06	4.5657536840E-05
3.1748682509E+00	1.3548946524E+00	1.8721614131E+00	7.7351749847E-06	3.4202668263E-06	4.5373227805E-06
1.8948244213E+01	1.0654054815E+00	1.2114431446E+01	4.5657536840E-05	4.5373227805E-06	2.8883619553E-05

POSITION EIGENVALUES OF ABOVE MATRIX

1	1.7907425310E+07
2	5.1968427911E+05
3	1.6131401489E+05

POSITION EIGENVECTORS OF ABOVE MATRIX

1	8.4876077739E-01	1.0235342610E-02	-5.2867795539E-01
2	5.0739180967E-02	9.9362263982E-01	1.0069550713E-01
3	5.2633703867E-01	-1.1229108336E-01	8.4282859131E-01

```
SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME    5.001 DAYS
                                                                                  PROBLEM. .      402     PAGE. .      113

**********************************************************************************************************************

                     FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                          THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  1.773E-06 X**2 +  1.963E-06 Y**2 +  4.443E-06 Z**2 + -6.161E-07 XY + -5.479E-06 XZ +  6.258E-07 YZ =   9

        XY HYPERELLIPSOID. . . . . .  1.773E-06 X**2 + -6.161E-07 XY +  1.963E-06 Y**2 =   9

        XZ HYPERELLIPSOID. . . . . .  1.773E-06 X**2 + -5.479E-06 XZ +  4.443E-06 Z**2 =   9

        YZ HYPERELLIPSOID. . . . . .  1.963E-06 Y**2 +  6.258E-07 YZ +  4.443E-06 Z**2 =   9


        VELOCITY EIGENVALUES OF ABOVE MATRIX
           1     1.0392591199E-04
           2     2.6393927543E-06
           3     7.8627514507E-07


        VELOCITY EIGENVECTORS OF ABOVE MATRIX
           1     8.4852786971E-01   -2.2362330069E-02   -5.2867795539E-01
           2     8.8854192891E-02    9.9094164674E-01    1.0069550713E-01
           3     5.2163721755E-01   -1.3241819718E-01    8.4282859131E-01

                     FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                          THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  3.626E+05 X**2 +  3.850E+05 Y**2 +  9.127E+05 Z**2 + -1.508E+05 XY + -1.123E+06 XZ +  1.173E+05 YZ =   9

        XY HYPERELLIPSOID. . . . . .  3.626E+05 X**2 + -1.508E+05 XY +  3.850E+05 Y**2 =   9

        XZ HYPERELLIPSOID. . . . . .  3.626E+05 X**2 + -1.123E+06 XZ +  9.127E+05 Z**2 =   9

        YZ HYPERELLIPSOID. . . . . .  3.850E+05 Y**2 +  1.173E+05 YZ +  9.127E+05 Z**2 =   9
```

SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME 5.001 DAYS

VEHICLE REACHED SPHERE OF INFLUENCE ON ORIGINAL NOMINAL TRAJECTORY AT TRAJECTORY TIME 204.417 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	-3.8548562082E+05	-3.1804362930E+05	2.6461169957E+05	5.6548233000E+05
VELOCITY RELATIVE TO TARGET PLANET	1.9742469517E+00	1.6000607476E+00	-1.3951539643E+00	2.8990170759E+00

B = 9.6813450458E+03 B DOT T = 4.4066604837E+03 B DOT R = 8.6203123654E+03

M MATRIX

6.2964049061E-01	-7.7688664076E-01	0.	1.2381857206E+05	-1.5277447361E+05	0.
-3.7387723089E-01	-3.0301491973E-01	-8.7658301070E-01	-7.3438662905E+04	-5.8455171882E+04	-1.7096169205E+05

VEHICLE REACHED SPHERE OF INFLUENCE ON MOST RECENT NOMINAL TRAJECTORY AT TRAJECTORY TIME 205.559 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	-4.5151009958E+05	-3.2485650143E+05	1.0186830310E+05	5.6548233192E+05
VELOCITY RELATIVE TO TARGET PLANET	1.9636848087E+00	1.5668592540E+00	-1.3915048395E+00	2.8718272350E+00

B = 1.8388513928E+05 B DOT T = -2.7961870987E+04 B DOT R = 1.8174674198E+05

VARIATION MATRIX

1.6991329230E+00	-5.7530035265E-02	5.4113114486E-02	5.4345369211E+06	-8.8618576460E+06	-3.1806674004E+04
-4.3911421383E+00	2.1353404270E+00	6.3708835095E-01	-2.5266352573E+07	-6.2088971905E+06	-4.6629220319E+06
-3.3634947613E-05	1.7165089957E-05	-5.1941024140E-06	-1.9819313893E+02	-4.6916932333E+01	-6.1545742210E+00

UNCERTAINTY IN TARGET CONDITIONS BEFORE CORRECTION

2.2530833633E+09	-1.1774206035E+10	-8.5926362987E+04
-1.1774206035E+10	6.9216609838E+10	5.0535988309E+05
-8.5926362987E+04	5.0535988309E+05	3.6903214915E+00

```
******************************************************************************************************************

EIGENVALUES OF ABOVE MATRIX
   1    2.4315183645E+08
   2    7.1226541369E+10
   3    6.1425645969E-04

EIGENVECTORS OF ABOVE MATRIX
   1    9.8574057481E-01   -1.6827215803E-01   -1.5473552544E-07
   2    1.6827215802E-01    9.8574057479E-01   -7.3274575599E-06
   3    1.3855361822E-06    7.1969345463E-06    9.9999999997E-01

          FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

4.036E-09 X**2 + 8.754E-08 Y**2 + 1.628E+03 Z**2 + 5.051E-09 XY + -5.038E-04 XZ + -2.386E-02 YZ =  9

   XY HYPERELLIPSOID. . . . . .  4.036E-09 X**2 + 5.051E-09 XY +  8.754E-08 Y**2 =  9

   XZ HYPERELLIPSOID. . . . . .  4.036E-09 X**2 + -5.038E-04 XZ + 1.628E+03 Z**2 =  9

   YZ HYPERELLIPSOID. . . . . .  8.754E-08 Y**2 + -2.386E-02 YZ + 1.628E+03 Z**2 =  9


GUIDANCE MATRIX -- THREE VARIABLE B-PLANE GUIDANCE POLICY
 -1.8700731899E-07   7.7394245370E-08  -3.3006760001E-08  -1.0000000000E+00   0.                 0.
  7.7165038126E-08   4.1027801052E-08  -1.5340728842E-08   0.                 -1.0000000000E+00  0.
 -3.1151940890E-08  -1.6055868350E-08   3.3590477900E-07   0.                  0.                -1.0000000000E+00
```

```
SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME    5.001 DAYS
                                                                  PROBLEM. .      402     PAGE. .     116

*****************************************************************************************************************

COVARIANCE MATRIX ASSOCIATED WITH VELOCITY COMPONENTS
    8.8208838733E-05    5.9139481091E-06    4.3643780900E-05
    5.9139481091E-06    2.9413242414E-06    2.8648481517E-06
    4.3643780900E-05    2.8648481517E-06    2.2376097198E-05


          EIGENVALUES OF ABOVE MATRIX
             1     1.1035946255E-04
             2     2.5421908234E-06
             3     6.2460680192E-07


          EIGENVECTORS OF ABOVE MATRIX
             1     8.9337536607E-01   -3.7515459175E-02   -4.4774216423E-01
             2     6.1057130369E-02    9.9740086116E-01    3.8256358849E-02
             3     4.4514321531E-01   -6.1515140284E-02    8.9334394573E-01

                 FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                      THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

 3.287E+05 X**2 +  3.937E+05 Y**2 +  1.281E+06 Z**2 + -8.330E+04 XY + -1.272E+06 XZ +  6.166E+04 YZ =   9

          XY HYPERELLIPSOID. . . . . .  3.287E+05 X**2 + -8.330E+04 XY +  3.937E+05 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  3.287E+05 X**2 + -1.272E+06 XZ +  1.281E+06 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  3.937E+05 Y**2 +  6.166E+04 YZ +  1.281E+06 Z**2 =   9
```

DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL TRAJECTORY

ESTIMATED	ACTUAL
-2.3862482253E+03	-2.3891669497E+03
-1.3687538284E+02	-1.2926306772E+02
-1.5464986656E+03	-1.5514388400E+03
-5.7509524418E-03	-5.7632663774E-03
-5.6464906343E-04	-5.4110233600E-04
-3.6826568159E-03	-3.6915652154E-03

COMMANDED CORRECTION	PERFECT CORRECTION
6.2376498681E-03	6.2512618351E-03
3.9862284886E-04	3.7523900037E-04
3.2397144402E-03	3.2469321131E-03

COMMANDED DELTA V. . . 7.0400941547E-03

ERROR IN CORRECTION DUE TO NAVIGATION UNCERTAINTY

1.3611966965E-05
-2.3383848490E-05
7.2176729419E-06

EXECUTION ERROR MATRIX

2.2119644329E-09	-1.7010600240E-11	-1.3824969490E-10
-1.7010600240E-11	2.4770592070E-09	-8.8349760564E-12
-1.3824969490E-10	-8.8349760564E-12	2.4063420735E-09

```
SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME    5.001 DAYS
                                                                PROBLEM. .      402     PAGE. .     118

*****************************************************************************************************

EIGENVALUES OF ABOVE MATRIX
   1     2.1390731427E-09
   2     2.4781462854E-09
   3     2.4781462854E-09


EIGENVECTORS OF ABOVE MATRIX
   1     8.8601796099E-01   -7.1694346929E-02   -4.5807433176E-01
   2     5.6621806484E-02    9.9730888543E-01   -4.6572074049E-02
   3     4.6018055569E-01    1.5326697920E-02    8.8769304858E-01

          FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

4.537E+08 X**2 . 4.037E+08 Y**2 + 4.171E+08 Z**2 + 6.418E+06 XY + 5.216E+07 XZ . 3.333E+06 YZ =  9

     XY HYPERELLIPSOID. . . . . .  4.537E+08 X**2 + 6.418E+06 XY + 4.037E+08 Y**2 =   9

     XZ HYPERELLIPSOID. . . . . .  4.537E+08 X**2 + 5.216E+07 XZ + 4.171E+08 Z**2 =   9

     YZ HYPERELLIPSOID. . . . . .  4.037E+08 Y**2 + 3.333E+06 YZ + 4.171E+08 Z**2 =   9


MODIFIED COVARIANCE MATRIX AT TIME OF GUIDANCE EVENT -- P(   5.001,   5.001)
   1.0787109435E+00  -1.0220070820E+00  -6.2464428639E-01   2.9727265173E-06  -2.8458835360E-06  -1.9851538116E-06
  -1.0220070820E+00   1.1989408415E+00   2.7643793165E-01  -2.7078798510E-06   3.1824296260E-06   9.9434446715E-07
  -6.2464428639E-01   2.7643793165E-01   7.9316682822E-01  -1.8671047420E-06   9.8531780575E-07   2.3636859043E-06
   2.9727265173E-06  -2.7078798510E-06  -1.8671047420E-06   2.2208957040E-09  -2.5099322687E-11  -1.4454980065E-10
  -2.8458835360E-06   3.1824296260E-06   9.8531780575E-07  -2.5099322687E-11   2.4862321267E-09  -5.2957374069E-12
  -1.9851538116E-06   9.9434446715E-07   2.3636859043E-06  -1.4454980065E-10  -5.2957374069E-12   2.4140769913E-09
```

```
*****************************************************************************************************************

          POSITION EIGENVALUES OF ABOVE MATRIX
             1     4.0452028200E-05
             2     2.4121856875E+00
             3     6.5859247375E-01


          POSITION EIGENVECTORS OF ABOVE MATRIX
             1     7.4194822326E-01    -6.6806296768E-01    -5.6610115843E-02
             2     5.4123906871E-01     6.4664493138E-01    -5.3750404950E-01
             3     3.9569319493E-01     3.6816056814E-01     8.4135883638E-01

                    FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                          THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

    1.361E+04 X**2 +  7.242E•03 Y**2 •  3.872E+03 Z**2 +  1.985E+04 XY +  1.451E+04 XZ +  1.059E+04 YZ =   9

          XY HYPERELLIPSOID. . . . . .  1.361E+04 X**2 •  1.985E•04 XY +  7.242E•03 Y**2 =   9

          XZ HYPERELLIPSOID. . . . . .  1.361E+04 X**2 +  1.451E+04 XZ •  3.872E+03 Z**2 =   9

          YZ HYPERELLIPSOID. . . . . .  7.242E+03 Y**2 •  1.059E+04 YZ •  3.872E+03 Z**2 =   9


          VELOCITY EIGENVALUES OF ABOVE MATRIX
             1     2.1418720057E-09
             2     2.4823305155E-09
             3     2.4970023008E-09


          VELOCITY EIGENVECTORS OF ABOVE MATRIX
             1     8.8036653981E-01     2.4274689386E-01    -4.0746619627E-01
             2     7.1377992308E-02     7.8150744877E-01     6.1979939474E-01
             3     4.6889224539E-01    -5.7473476755E-01     6.7068622259E-01
```

```
SIMULATION MODE -- GUIDANCE EVENT AT TRAJECTORY TIME    5.001 DAYS
                                                              PROBLEM. .      402    PAGE. .     120

**********************************************************************************************************

                  FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                       THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  4.521E+08 X**2 +  4.023E+08 Y**2 +  4.159E+08 Z**2 +  9.243E+06 XY +  5.416E+07 XZ +  2.318E+06 YZ =   9

        XY HYPERELLIPSOID. . . . . .  4.521E+08 X**2 +  9.243E+06 XY +  4.023E+08 Y**2 =   9

        XZ HYPERELLIPSOID. . . . . .  4.521E+08 X**2 +  5.416E+07 XZ +  4.159E+08 Z**2 =   9

        YZ HYPERELLIPSOID. . . . . .  4.023E+08 Y**2 +  2.318E+06 YZ +  4.159E+08 Z**2 =   9


UNCERTAINTY IN TARGET CONDITION AFTER CORRECTION
      2.6345634897E+05   -1.6962403807E+05   -1.3948597845E+00
     -1.6962403807E+05    1.5245861337E+06    1.1702385369E+01
     -1,3948597845E+00    1.1702385369E+01    9.2021806620E-05


        EIGENVALUES OF ABOVE MATRIX
           1     2.4104007945E+05
           2     1.5470024033E+06
           3     2.1616312124E-06


        EIGENVECTORS OF ABOVE MATRIX
           1     9.9138057241E-01   -1.3101358958E-01    3.7967814424E-07
           2     1.3101358958E-01    9.9138057238E-01   -7.6335357978E-06
           3     6.2369139007E-07    7.6174820853E-06    9.9999999997E-01
```

```
**************************************************************************************************************

                    FOR THE NORMAL DISTRIBUTION X = N(0,Q) AND THE 3 SIGMA LEVEL
                         THE HYPERELLIPSOID HAS THE FOLLOWING EQUATION

  4.155E-06 X**2 +  2.766E-05 Y**2 +  4.626E+05 Z**2 + -1.772E-06 XY +  3.513E-01 XZ + -7.063E+00 YZ =   9

        XY HYPERELLIPSOID. . . . . .  4.155E-06 X**2 + -1.772E-06 XY +  2.766E-05 Y**2 =   9

        XZ HYPERELLIPSOID. . . . . .  4.155E-06 X**2 +  3.513E-01 XZ +  4.626E+05 Z**2 =   9

        YZ HYPERELLIPSOID. . . . . .  2.766E-05 Y**2 + -7.063E+00 YZ +  4.626E+05 Z**2 =   9


ACTUAL ERROR IN CORRECTION         ACTUAL CORRECTION
      1.4670325164E-05              6.2523201933E-03
      3.5169977199E-06              4.0213984658E-04
     -3.8698822204E-05              3.2010156180E-03


ERROR AT TARGET CONDITIONS
DUE TO NAVIGATION UNCERTAINTY    DUE TO EXECUTION ERROR

     -2.8096950341E+02              4.9790171437E+01
      2.3239229154E+02             -2.1205269456E+02
      1.6451217264E-03             -2.8343897640E-03


ACTUAL ERROR AT TARGET AFTER CORRECTION

     -2.3117933197E+02
      2.0339596976E+01
     -1.1892680376E-03
```

****

ORIGINAL NOMINAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT TRAJECTORY TIME 204.41694 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	-3.8542997390E+05	-3.1805911324E+05	2.6467414164E+05	5.5548233000E+05
VELOCITY RELATIVE TO TARGET PLANET	1.9742494809E+00	1.6000764286E+00	-1.3951539811E+00	2.8990274674E+00

B = 9.6376361394E+03 B DOT T = 4.4520581977E+03 B DOT R = 8.5477019227E+03

****

MOST RECENT NOMINAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT TRAJECTORY TIME 204.41693 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	-3.8546333616E+05	-3.1806877218E+05	2.6461394178E+05	5.6548233000E+05
VELOCITY RELATIVE TO TARGET PLANET	1.9742740629E+00	1.6000067783E+00	-1.3947395380E+00	2.8988063311E+00

B = 9.6373559610E+03 B DOT T = 4.4522470243E+03 B DOT R = 8.5472876606E+03

****

ACTUAL TRAJECTORY ENCOUNTERED SPHERE OF INFLUENCE AT TRAJECTORY TIME 204.41697 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	-3.8546996417E+05	-3.1807978844E+05	2.6459104380E+05	5.6548233000E+05
VELOCITY RELATIVE TO TARGET PLANET	1.9742677995E+00	1.5999977267E+00	-1.3947335710E+00	2.8987941982E+00

B = 9.6628297878E+03 B DOT T = 4.4572866203E+03 B DOT R = 8.5733818002E+03

****

****

ORIGINAL NOMINAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO TARGET PLANET AT 206.61807 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	1.7309593393E+03	-1.7144577368E+03	-5.2550890939E+03	5.7923697180E+03
VELOCITY RELATIVE TO TARGET PLANET	2.9484036248E+00	3.7827485397E+00	-2.7836158379E-01	4.8041394256E+00

****

****

MOST RECENT NOMINAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO TARGET PLANET AT 296.618 DAYS

	X	Y	Z	RESULTANT
POSITION RELATIVE TO TARGET PLANET	1.7563977087E+03	-1.6818656763E+03	-5.2566942612E+03	5.7919288342E+03
VELOCITY RELATIVE TO TARGET PLANET	2.9451450575E+00	3.7860323919E+00	-2.6773991264E-01	4.8041237852E+00

****

```
********************************************************************************************************************

     ACTUAL TRAJECTORY REACHED POINT OF CLOSEST APPROACH TO TARGET PLANET AT  206.61836 DAYS

                                              X                  Y                  Z               RESULTANT
     POSITION RELATIVE TO TARGET PLANET   1.7436771504E+03  -1.7072084153E+03  -5.2772150247E+03  5.8141180754E+03
     VELOCITY RELATIVE TO TARGET PLANET   2.9460719843E+00   3.7772023193E+00  -2.7630031164E-01  4.7982225209E+00

********************************************************************************************************************
```

```
S U M MA R Y   O F   S I M U L A T I O N   M O D E
                                                                                    PROBLEM. .        402     PAGE. .     528

   VIRTUAL MASS TRAJECTORY

     ACCURACY USED IN TRAJECTORY

         NOMINAL          ACTUAL
       5.000E-06        5.000E-06

     BODIES CONSIDERED IN TRAJECTORY

       NOMINAL          ACTUAL

       SUN              SUN
       EARTH            EARTH
       MARS             MARS
       JUPITER          JUPITER
       MOON             MOON

     GRAVITATIONAL CONSTANT BIASES USED IN ACTUAL TRAJECTORY

       SUN          0.
       MARS         0.

     EPHEMERIS BIASES USED IN ACTUAL TRAJECTORY

       SEMI-MAJOR AXIS   0.
       ECCENTRICITY      0.
       INCLINATION       0.

     INITIAL TRAJECTORY TIME       0.       DAYS. JULIAN DATE  2441887.8929124624     CALENDAR DATE  7 24   9 25 47.637, 1973
     FINAL TRAJECTORY TIME     210.60709 DAYS. JULIAN DATE  4856907.8929124475     CALENDAR DATE  2 20   0  0  0.    , 1974
```

AT INITIAL TIME, ECLIPTIC COORDINATES OF VEHICLE

	X	Y	Z	RESULTANT
RELATIVE TO SUN				
POSITION	7.9030349297E+07	-1.2979967127E+08	7.5601537160E+01	1.5196628169E+08
VELOCITY	3.4316535000E+01	1.2503124000E+01	6.2151790000E+00	3.7048362098E+01
RELATIVE TO EARTH				
POSITION	-1.1319096937E+03	-6.5756145287E+03	7.5601537160E+01	6.6727536727E+03
VELOCITY	9.3594827098E+00	-2.8817410726E+00	6.2151790000E+00	1.1598827450E+01
RELATIVE TO MARS				
POSITION	9.7490552553E+07	-3.6565244075E+08	-5.4137295877E+06	3.7846456074E+08
VELOCITY	5.7534909782E+01	1.2345845297E+01	5.6428625055E+00	5.9114529829E+01

AT FINAL TIME

	X	Y	Z	RESULTANT
ORIGINAL NOMINAL TRAJECTORY				
RELATIVE TO SUN				
POSITION	-1.8048991683E+07	2.3669934969E+08	5.7873731161E+06	2.3745703176E+08
VELOCITY	-2.2061953129E+01	2.5713042224E+00	1.6604848082E+00	2.2273270776E+01
RELATIVE TO EARTH				
POSITION	1.1128566205E+08	1.6492170933E+08	5.7873731161E+06	1.9904060509E+08
VELOCITY	-7.1204088811E+00	2.8725725184E+01	1.6604848082E+00	2.9641604541E+01
RELATIVE TO MARS				
POSITION	4.1121157284E+05	8.4658021153E+05	3.7356792683E+05	1.0125937034E+06
VELOCITY	1.1564216525E+00	2.4140255193E+00	1.0881683137E+00	2.8894533263E+00
MOST RECENT NOMINAL TRAJECTORY				
RELATIVE TO SUN				
POSITION	-1.8049157368E+07	2.3669921768E+08	5.7875480731E+06	2.3745691703E+08
VELOCITY	-2.2062388672E+01	2.5710079421E+00	1.6610180546E+00	2.2273707745E+01
RELATIVE TO EARTH				
POSITION	1.1128549636E+08	1.6492157732E+08	5.7875480731E+06	1.9904040816E+08
VELOCITY	-7.1208444234E+00	2.8725428903E+01	1.6610180546E+00	2.9641451921E+01
RELATIVE TO MARS				
POSITION	4.1104588798E+05	8.4644820112E+05	3.7374288380E+05	1.0124806282E+06
VELOCITY	1.1559861102E+00	2.4137292390E+00	1.0887015601E+00	2.8892323917E+00

SUMMARY OF SIMULATION MODE

PROBLEM. . 402 PAGE. . 530

AT FINAL TIME

	X	Y	Z	RESULTANT
ACTUAL TRAJECTORY				
RELATIVE TO SUN				
POSITION	-1.8047451149E+07	2.3669903944E+08	5.7859741896E+06	2.3745657132E+08
VELOCITY	-2.2057493751E+01	2.5706069270E+00	1.6566495464E+00	2.2268487561E+01
RELATIVE TO EARTH				
POSITION	1.1128720258E+08	1.6492139908E+08	5.7859741896E+06	1.9904116868E+08
VELOCITY	-7.1159495024E+00	2.8725027888E+01	1.6566495464E+00	2.9639643254E+01
RELATIVE TO MARS				
POSITION	4.1275210767E+05	8.4626995679E+05	3.7216900029E+05	1.0124460020E+06
VELOCITY	1.1608810312E+00	2.4133282239E+00	1.0843330519E+00	2.8892172041E+00

AT CLOSEST APPROACH TO THE TARGET PLANET

	X	Y	Z	RESULTANT
ORIGINAL NOMINAL TRAJECTORY AT TRAJECTORY TIME 206.618 DAYS				
POSITION	1.7309593393E+03	-1.7144577368E+03	-5.2550890939E+03	5.7923497180E+03
VELOCITY	2.9451450575E+00	3.7860323919E+00	-2.6773991264E-01	4.8041394256E+00
MOST RECENT NOMINAL TRAJECTORY AT TRAJECTORY TIME 206.618 DAYS				
POSITION	1.7563977087E+03	-1.6818656763E+03	-5.2566942612E+03	5.7919288342E+03
VELOCITY	2.9451450575E+00	3.7860323919E+00	-2.6773991264E-01	4.8041237852E+00
ACTUAL TRAJECTORY AT TRAJECTORY TIME 206.618 DAYS				
POSITION	1.7436771504E+03	-1.7072084153E+03	-5.2772150247E+03	5.8141180754E+03
VELOCITY	2.9460719843E+00	3.7772023193E+00	-2.7630031164E-01	4.7982225209E+00

INFORMATION AT SPHERE OF INFLUENCE

	X	Y	Z	RESULTANT
ORIGINAL NOMINAL TRAJECTORY				
POSITION	-3.8542997390E+05	-3.1805911324E+05	2.6467414164E+05	5.6548233000E+05
VELOCITY	-3.8546333616E+05	-3.1806877218E+05	2.6461394178E+05	5.6548233000E+05

B = 9.6376361394E+03, B DOT T = 4.4520581977E+03, B DOT R = 8.5477019227E+03

TRAJECTORY TIME 204.417 DAYS

	X	Y	Z	RESULTANT
MOST RECENT NOMINAL TRAJECTORY				
POSITION	-3.8546333616E+05	-3.1806877218E+05	2.6461394178E+05	6.0595751108E+05
VELOCITY	-3.8546996417E+05	-3.1807978844E+05	2.6459104380E+05	5.6548233000E+05

B = 9.6373559610E+03, B DOT T = 4.4522470243E+03, B DOT R = 8.5472876606E+03

TRAJECTORY TIME 204.417 DAYS

	X	Y	Z	RESULTANT
ACTUAL TRAJECTORY				
POSITION	-3.8546996417E+05	-3.1807978844E+05	2.6459104380E+05	5.6548233000E+05
VELOCITY	1.9742494899E+00	1.6000764286E+00	-1.3951539811E+00	2.8990274674E+00

B = 9.6628297878E+03, B DOT T = 4.4572866203E+03, B DOT R = 8.5733818002E+03

TRAJECTORY TIME 204.417 DAYS

****

MISCELLANEOUS INFORMATION USED IN SIMULATION MODE

THE STATE TRANSITION MATRIX WAS COMPUTED ANALYTICALLY FROM THE PATCHED-CONIC TECHNIQUE EXCEPT FOR THE FOLLOWING CONDITION

IF THE TIME INTERVAL OVER WHICH THE STATE-TRANSITION MATRIX WAS COMPUTED WAS GREATER THAN 8.000 DAYS
THE GOVERNING BODY WAS ASSUMED TO BE THE SUN IN THE ANALYTICAL CALCULATION

SUMMARY OF SIMULATION MODE

NUMBER OF MEASUREMENTS TAKEN 71

TOTAL NUMBER OF EVENTS 17
EIGENVECTOR EVENTS. 4
PREDICTION EVENTS 5
GUIDANCE EVENTS 3
QUASI-LINEAR FILTERING EVENTS . 5

VARIANCES OF ERRORS USED IN GUIDANCE EVENTS

RESOLUTION	PROPORTIONALITY	POINTING ANGLE 1	POINTING ANGLE 2
9.00000000E-10	2.50000000E-05	5.00000000E-05	5.00000000E-05

ACTUAL ERRORS USED IN GUIDANCE EVENT

	RESOLUTION	PROPORTIONALITY	POINTING ANGLE 1	POINTING ANGLE 2
1	1.50000000E-05	-2.50000000E-03	-5.40000000E-03	6.00000000E-03
2	-1.80000000E-05	4.00000000E-03	2.00000000E-03	-4.00000000E-03
3	1.10000000E-05	2.00000000E-03	7.00000000E-03	2.00000000E-03

STATION LOCATION CONSTANTS

	ALTITUDE	LATITUDE	LONGITUDE
STATION 1	1.7994344588E-02	3.5384000000E+01	-2.0391205250E+00
STATION 2	8.7266462600E-04	4.0417000000E+01	-6.4001223671E-02
STATION 3	8.7266462600E-04	-3.5311000000E+01	2.6029142333E+00

THE DYNAMIC NOISE MATRIX IS A DIAGONAL MATRIX WHERE THE DIAGONAL IS COMPUTED FROM THE FOLLOWING CONSTANTS

X	Y	Z
1.0000000000E-22	1.0000000000E-22	1.0000000000E-22

```
ACTUAL UNMODELLED ACCELERATION (ACTUAL DYNAMIC NOISE)
                                                      X            Y            Z
     FROM   0.    DAYS THROUGH  210.607 DAYS . . .    0.           0.           0.

MEASUREMENT NOISE WAS CONSTANT
     RANGE (EARTH-CENTERED. . . . .      1.0000000000E-06
     RANGE-RATE (EARTH-CENTERED . .      1.0000000000E-12
     RANGE (STATION NUMBER 1) . . .      2.5000000000E-05
     RANGE-RATE (STATION NUMBER 1).      9.0000000000E-12
     RANGE (STATION NUMBER 2) . . .      2.5000000000E-05
     RANGE-RATE (STATION NUMBER 2).      9.0000000000E-12
     RANGE (STATION NUMBER 3) . . .      2.5000000000E-05
     RANGE-RATE (STATION NUMBER 3).      9.0000000000E-12
     STAR PLANET ANGLE NUMBER 1 . .      2.5000000000E-09
     STAR PLANET ANGLE NUMBER 2 . .      2.5000000000E-09
     STAR PLANET ANGLE NUMBER 3 . .      2.5000000000E-09
     APPARENT PLANET DIAMETER . . .      2.5000000000E-09

THE UNCERTAINTIES IN THE ACTUAL MEASUREMENTS ARE COMPUTED FROM THE FOLLOWING VARIANCES
     RANGE (EARTH-CENTERED. . . . .      0.
     RANGE-RATE (EARTH-CENTERED . .      0.
     RANGE (STATION NUMBER 1) . . .      2.5000000000E-07
     RANGE-RATE (STATION NUMBER 1).      9.0000000000E-14
     RANGE (STATION NUMBER 2) . . .      2.5000000000E-07
     RANGE-RATE (STATION NUMBER 2).      9.0000000000E-14
     RANGE (STATION NUMBER 3) . . .      2.5000000000E-07
     RANGE-RATE (STATION NUMBER 3).      9.0000000000E-14
     STAR PLANET ANGLE NUMBER 1 . .      0.
     STAR PLANET ANGLE NUMBER 2 . .      0.
     STAR PLANET ANGLE NUMBER 3 . .      0.
     APPARENT PLANET DIAMETER . . .      0.

DIRECTION COSINES FOR THREE STAR PLANET ANGLES
     1   -6.1351000000E-02    2.3788600000E-01   -9.6935500000E-01
     2    2.8986000000E-02    9.6038800000E-01   -2.7714100000E-01
     3    2.0196300000E-01    8.3134300000E-01   -5.1778400000E-01
```

S U M MA R Y O F S I M U L A T I O N M O D E

STATISTICAL DATA FOR SIMULATION MODE

INITIAL STATE VECTOR

	NOMINAL	ACTUAL
RX	7.9030349297E+07	7.9030349797E+07
RY	-1.2979967127E+08	-1.2979967177E+08
RZ	7.5601537160E+01	7.6101537160E+01
VX	3.4316535000E+01	3.4315035000E+01
VY	1.2503124000E+01	1.2504624000E+01
VZ	6.2151790000E+00	6.2136790000E+00

FINAL STATE VECTOR

	ORIGINAL NOMINAL	MOST RECENT NOMINAL	ACTUAL
RX	-1.8048991683E+07	-1.8049157368E+07	-1.8047451149E+07
RY	2.3669934969E+08	2.3669921768E+08	2.3669903944E+08
RZ	5.7873731161E+06	5.7875480731E+06	5.7859741896E+06
VX	-2.2061953129E+01	-2.2062388672E+01	-2.2057493751E+01
VY	2.5713042224E+00	2.5710079421E+00	2.5706069270E+00
VZ	1.6604848082E+00	1.6610180546E+00	1.6566495464E+00

DEVIATION OF THE STATE VECTOR FROM THE MOST RECENT NOMINAL TRAJECTORY AT FINAL TIME

ESTIMATED	ACTUAL
1.6254778338E+03	1.7062196850E+03
-1.1874335621E+02	-1.7824432659E+02
-1.6139933769E+03	-1.5738835061E+03
4.6648415367E-03	4.8949209810E-03
-2.3033383527E-04	-4.0101509414E-04
-4.4862035605E-03	-4.3685081921E-03

DEVIATION OF STATE VECTOR FROM ORIGINAL NOMINAL AT FINAL TIME

ESTIMATED	ACTUAL
1.4597929767E+03	1.5405348278E+03
-2.5075376843E+02	-3.1025473881E+02
-1.4390364060E+03	-1.3989265352E+03
4.2292992785E-03	4.4593787228E-03
-5.2661414319E-04	-6.9729540206E-04
-3.9529571868E-03	-3.8352618184E-03

ACTUAL ORBIT DETERMINATION INACCURACY AT FINAL TIME

8.0741851153E+01
-5.9500970379E+01
4.0109870807E+01
2.3007944432E-04
-1.7068125887E-04
1.1769536839E-04

INITIAL COVARIANCE MATRIX

1.00000000E+00	0.	0.	0.	0.	0.
0.	1.00000000E+00	0.	0.	0.	0.
0.	0.	1.00000000E+00	0.	0.	0.
0.	0.	0.	9.00000000E-06	0.	0.
0.	0.	0.	0.	9.00000000E-06	0.
0.	0.	0.	0.	0.	9.00000000E-06

FINAL COVARIANCE MATRIX

2.61138523E+07	-1.90847822E+07	1.42580921E+07	7.49800573E+01	-5.47723829E+01	4.09745308E+01
-1.90847822E+07	1.39477389E+07	-1.04202394E+07	-5.47976706E+01	4.00293056E+01	-2.99454180E+01
1.42580921E+07	-1.04202394E+07	7.78497966E+06	4.09389063E+01	-2.99055830E+01	2.23722587E+01
7.49800573E+01	-5.47976706E+01	4.09389063E+01	2.15288398E-04	-1.57266595E-04	1.17649161E-04
-5.47723829E+01	4.00293056E+01	-2.99055830E+01	-1.57266595E-04	1.14882123E-04	-8.59418864E-05

SUMMARY OF SIMULATION MODE

PROBLEM. = 402 PAGE. = 536

4.09745308E+01	-2.99454180E+01	2.23722587E+01	1.17649161E-04	-8.59418864E-05	6.42927937E-05

Martin Marietta Corporation
Denver, Colorado
August 15, 1969

IX. REFERENCES

1. Joseph, A.E.; and Richard, R. J.: Space Research Conic Program. Engineering Planning Document 406, Jet Propulsion Laboratory, July 1966.

2. Danby, J. M. A.: Matrizant of Keplerian Motion. AIAA J., -3, no. 4, Apr. 1965.